RX8/RX11
floppy disk system
maintenance manual

EK-RX01-MM-002

digital equipment corporation · maynard. massachusetts

1st Edition, May 1975
2nd Printing (Rev), September 1975
3rd Printing, July 1976
4th Printing (Rev), December 1976

Printed in U.S.A.

CONTENTS

CONTENTS (Cont)

CONTENTS (Cont)

CONTENTS (Cont)

ILLUSTRATIONS

ILLUSTRATIONS (Cont)

ILLUSTRATIONS (Cont)

TABLES

CHAPTER 1
GENERAL INFORMATION

This manual presents information on the installation, operation, programming, theory of operation, and maintenance of the RX8 or RX11 Floppy Disk System. Chapter 2 (Installation and Operation) should be consulted for unpacking and installation information. Chapter 2 also provides information on the proper care of the media and should be read carefully.

1.1 INTRODUCTION

The RX8 and RX11 Floppy Disk Systems consist of an RX01 subsystem and either an RX8E interface for a PDP-8 system or an RX11 interface for a PDP-11 system.

The RX01 is a low cost, random access, mass memory device that stores data in fixed length blocks on a preformatted, IBM-compatible, flexible diskette. Each drive can store and retrieve up to 256K 8-bit bytes of data (PDP-11 or PDP-8) or 128K 12-bit words (PDP-8). The RX01 consists of one or two flexible disk drives, a single read/write electronics module, a microprogrammed controller module, and a power supply, enclosed in a rack-mountable, 10-1/2 inch, self-cooled chassis. A cable is included for connection to either a PDP-8 interface module for use on the PDP-8 Omnibus or a PDP-11 interface for use on the PDP-11 Unibus.

The RX01 performs implied seeks. Given an absolute sector address, the RX01 locates the desired sector and performs the indicated function, including automatic head position verification and hardware calculation and verification of the Cyclic Redundancy Check (CRC) character. The CRC character that is read and generated is compatible with IBM 3740 equipment.

The RX01 connects to the M8357 Omnibus interface module, which converts the RX01 I/O bus to a PDP-8 family Omnibus structure. It controls interrupts to the CPU initiated by the RX01, controls data interchange between the RX01 and the host CPU, and handles I/O transfers used to test status conditions.

The RX01 connects to the M7846 Unibus interface module, which converts the RX01 I/O bus to a PDP-11 Unibus structure. It controls interrupts to the CPU initiated by the RX01, decodes Unibus addresses for register selection, and handles data interchange between the RX01 and the host CPU.

The interface modules are dc powered by their host processor.

1.2 PHYSICAL DESCRIPTION

A complete system consists of the following components:

M7726 controller module
M7727 read/write electronics module
H771A or B power supply
RX01-CA floppy disk drive (60 Hz, max of 2)
RX01-CC floppy disk drive (50 Hz, max of 2)
M8357 (RX8E) or M7846 (RX11) interfaces

All components except the interface are housed in a 10-1/2 in. rack-mountable box. The power supply, M7726 module, and M7727 module are mounted above the drives. Interconnection from the RX01 to the interface is with a 40-conductor BC05L-15 cable of standard length (15 ft). Figure 1-1 is a configuration drawing of the system, and Figure 1-2 is a front view of a dual drive system.

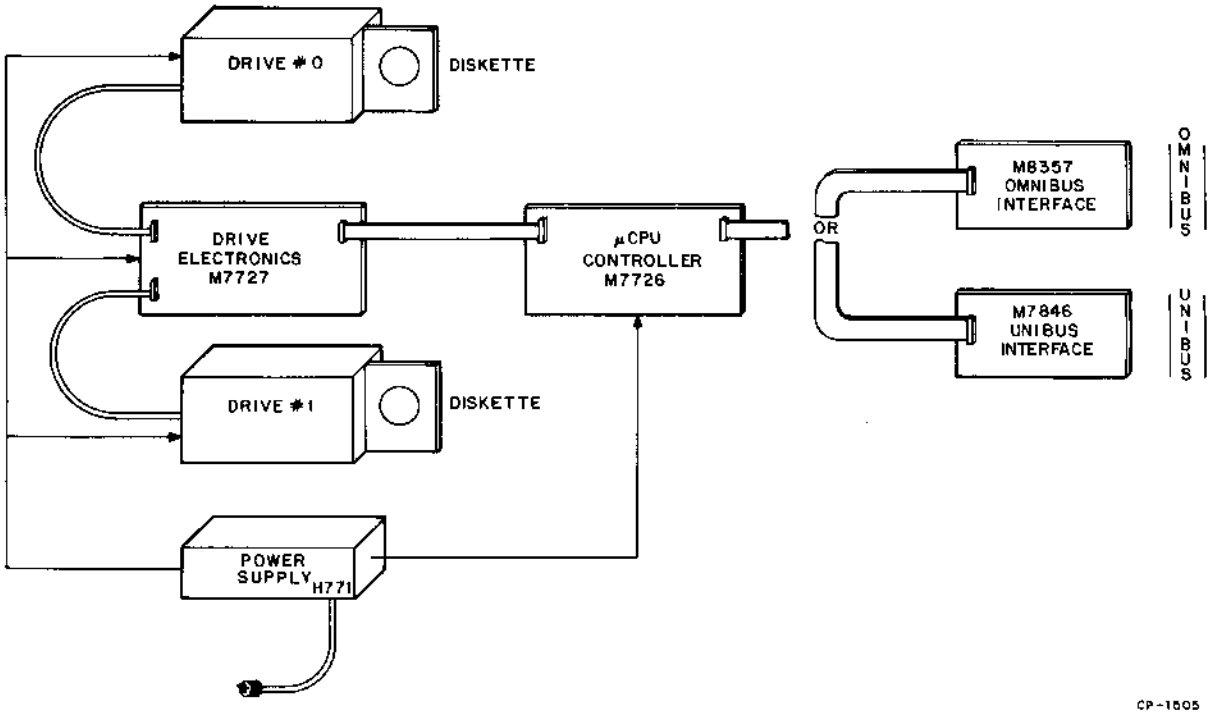


Figure 1-1 Floppy Disk System Configuration

1.2.1 RX8E/RX11 Interfaces

Interface modules M8357 (RX8E) and M7846 (RX11) are both quad modules. The M8357 plugs into an Omnibus slot and allows the RX01 to be used on the PDP-8 processors. The M7846 plugs into an SPC (small peripheral controller) slot with any PDP-11 processor. Figure 1-3 shows the M8357 module and its major sections. Figure 1-4 shows the M7846 module and its major sections.

1.2.2 Microprogrammed Controller

The M7726 microprogrammed controller module is located in the RX01 cabinet as shown in Figure 1-5. The M7726 is hinged on the left side and lifts up for access to the M7727 read/write electronics module.

1.2.3 Read/Write Electronics

The M7727 read/write electronics module is located in the RX01 cabinet as shown in Figure 1-5.

1.2.4 Electro-Mechanical Drive

A maximum of two drives can be attached to the read/write electronics. The electro-mechanical drives are mounted side by side under the read/write electronics board (M7727). Figure 1-6, which is an underside view of the drive, shows the drive motor connected to the spindle by a belt. (This belt and the small pulley are different on the 50 Hz and 60 Hz units; see Paragraph 2.2.3.2 for complete input power modification requirements.) Figure 1-7 is the top view showing the electro-mechanical components of the drive.

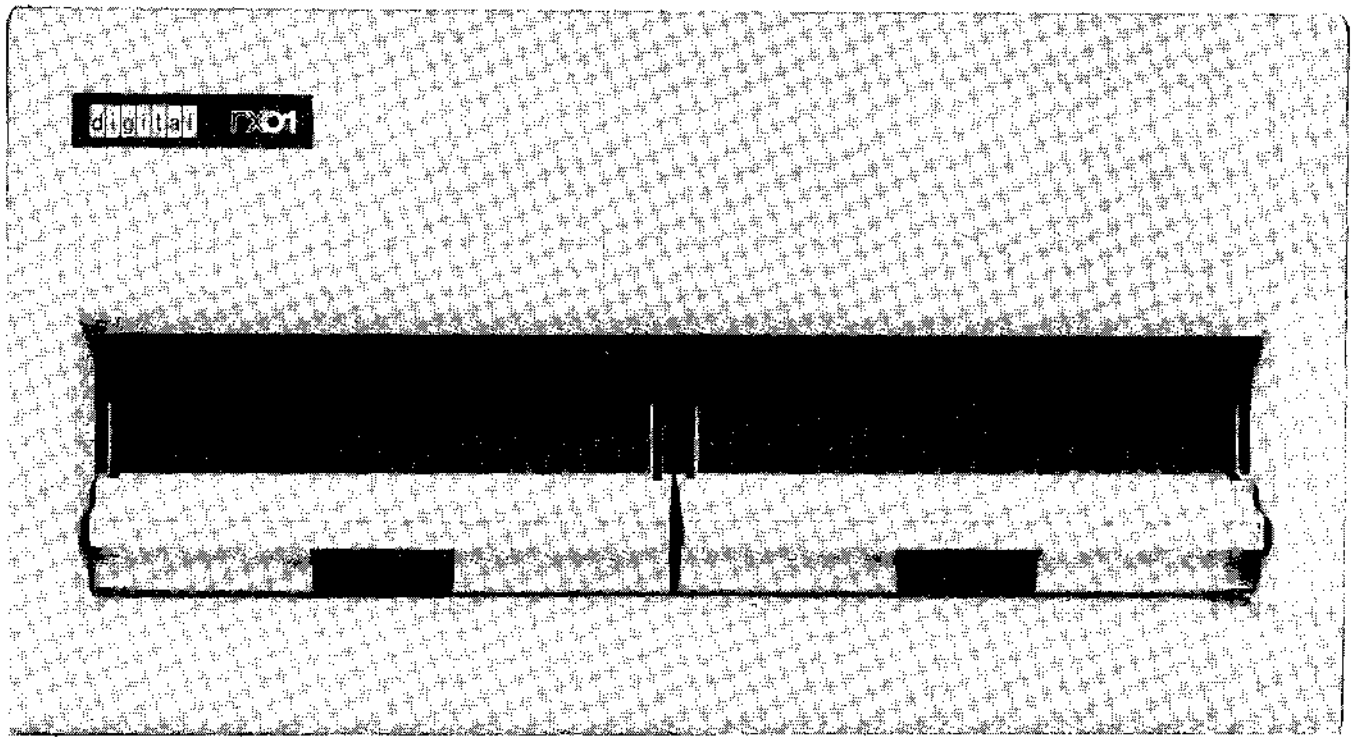


Figure 1-2 Front View of the Floppy Disk System

1.2.5 Power Supply

The H771 power supply is mounted at the rear of the RX01 cabinet as shown in Figure 1-5. The H771A is rated at 60 Hz ± 1/2 Hz over a voltage range of 90–132 Vac. The H771C and D are rated at 50 Hz ± 1/2 Hz over four voltage ranges:

90–120 Vac, 100–132 Vac } 3.5 A circuit breaker; H771C

180–240 Vac, 200–264 Vac } 1.75 A circuit breaker; H771D

Two power harnesses are provided to adapt the H771C or D to each voltage range. This is not applicable to the H771A. See Paragraph 2.2.3.2 for complete input power modification requirements.

1.3 SYSTEMS COMPATIBILITY

This section describes the physical, electrical, and logical aspects of IBM compatibility as defined for data interchange with IBM system 3740 devices.

1.3.1 Media

The media used on the RX8 or RX11 Floppy Disk System is compatible with the IBM 3740 family of equipment and is shown in Figure 1-8.

The "diskette" media was designed by applying tape technology to disk architecture. This resulted in a flexible oxide-on-mylar surface encased in a plastic envelope with a hole for the read/write head, a hole for the drive spindle hub, and a hole for the hard index mark. The envelope is lined with a fiber material that cleans the diskette surface. The media is supplied to the customer preformatted and pretested.

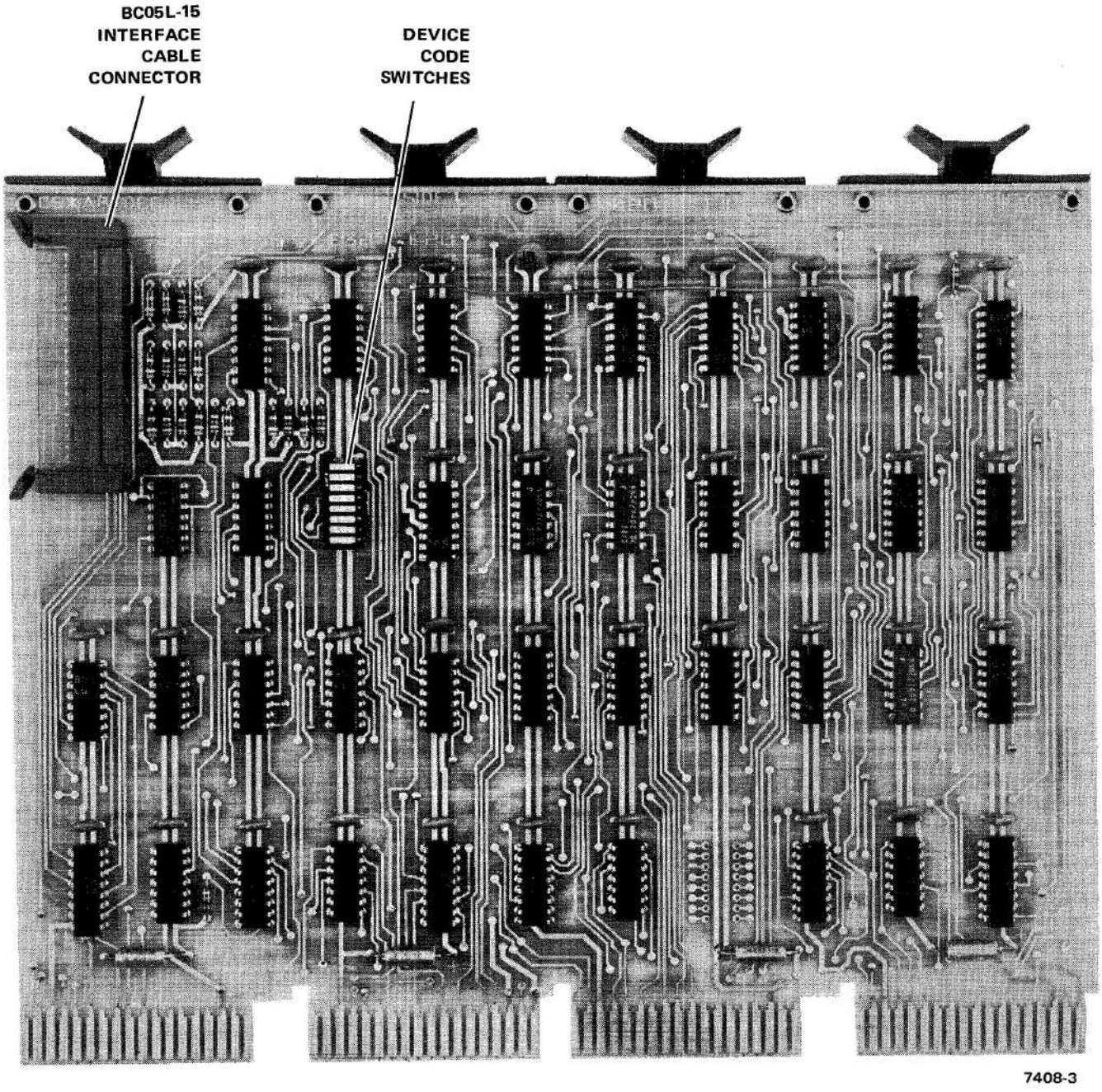


Figure 1-3 M8357 Module (RX8E Interface)

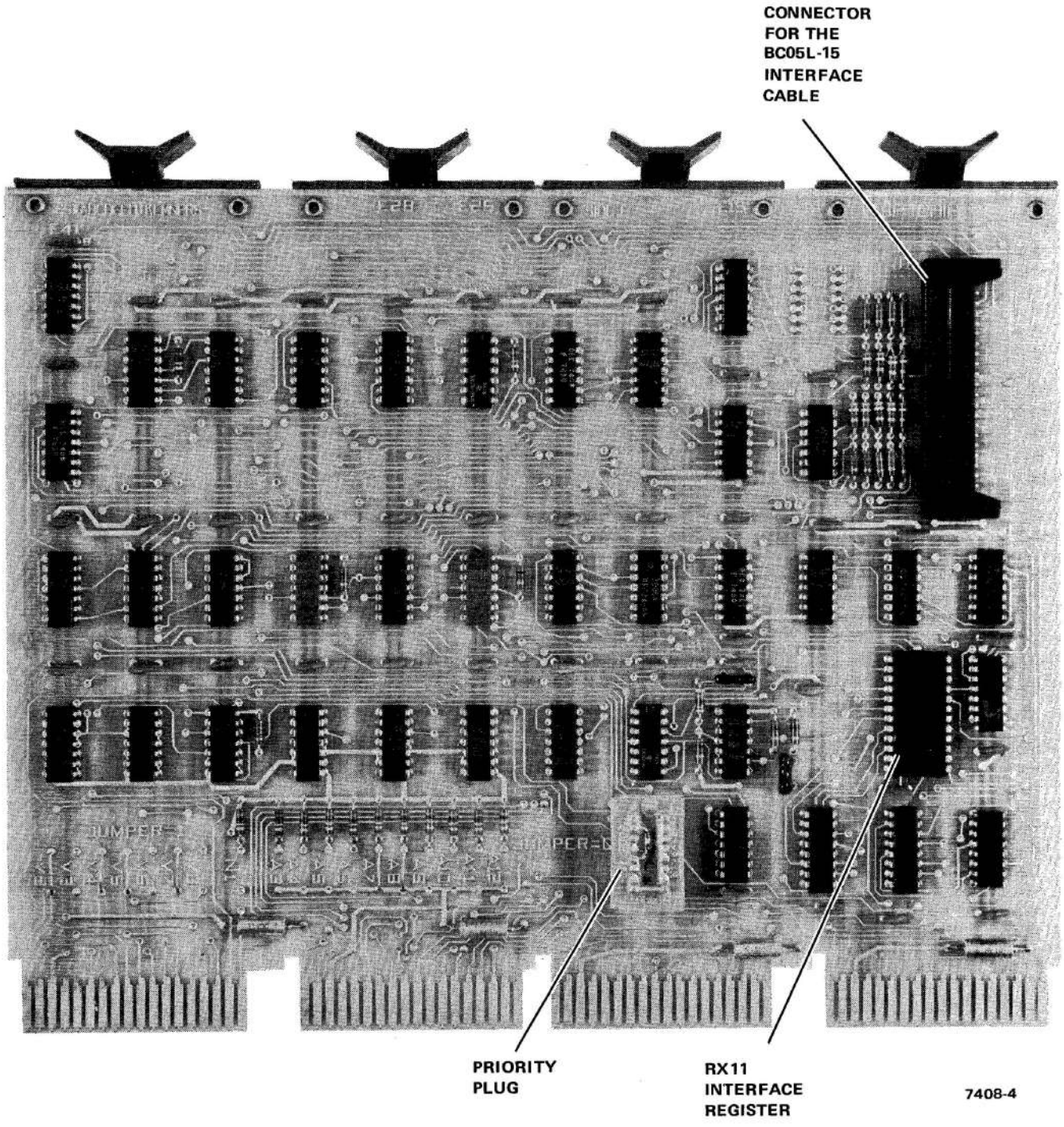


Figure 1-4 M7846 Module (RX11 Interface)

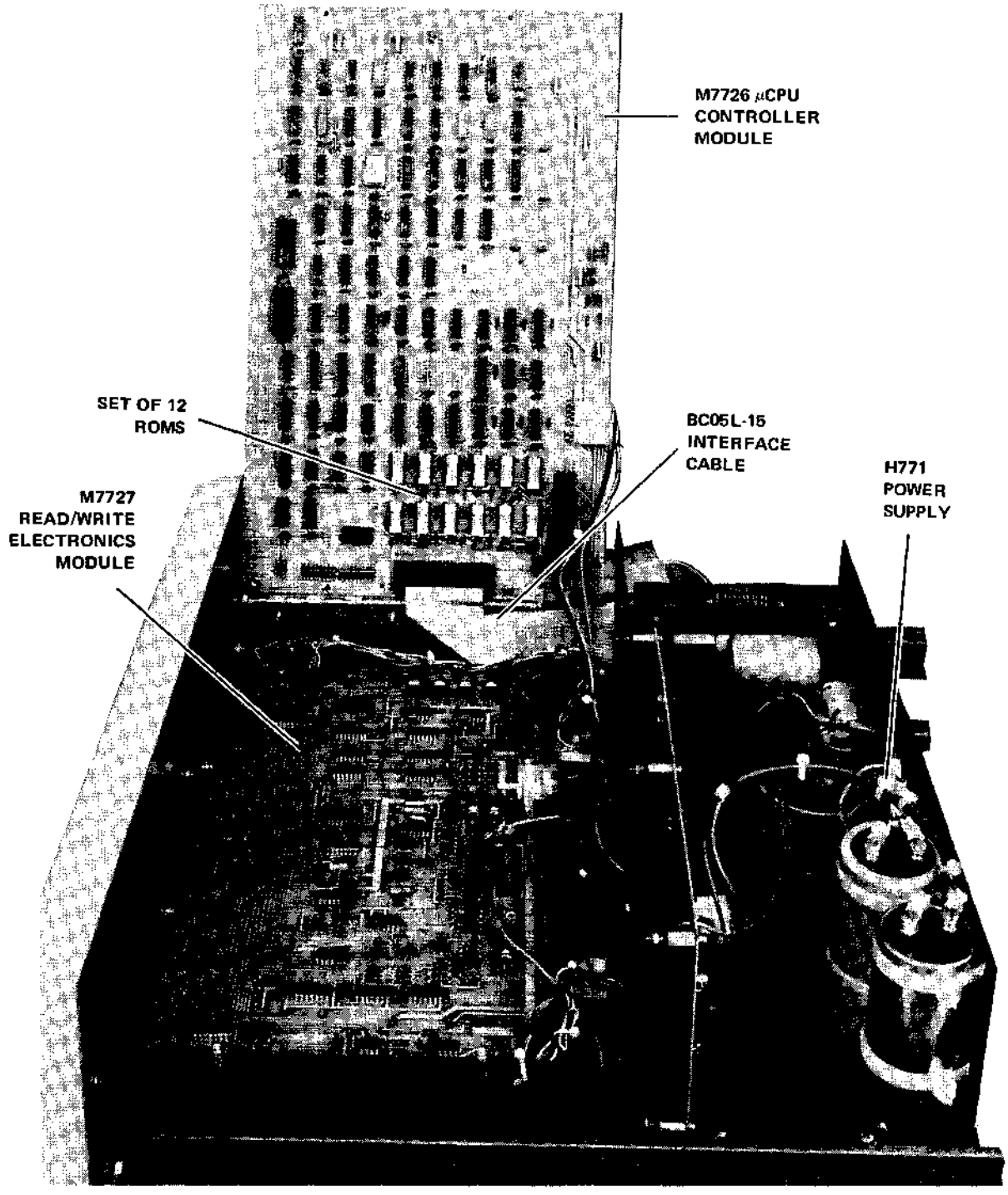


Figure 1-5 Top View of the RX01

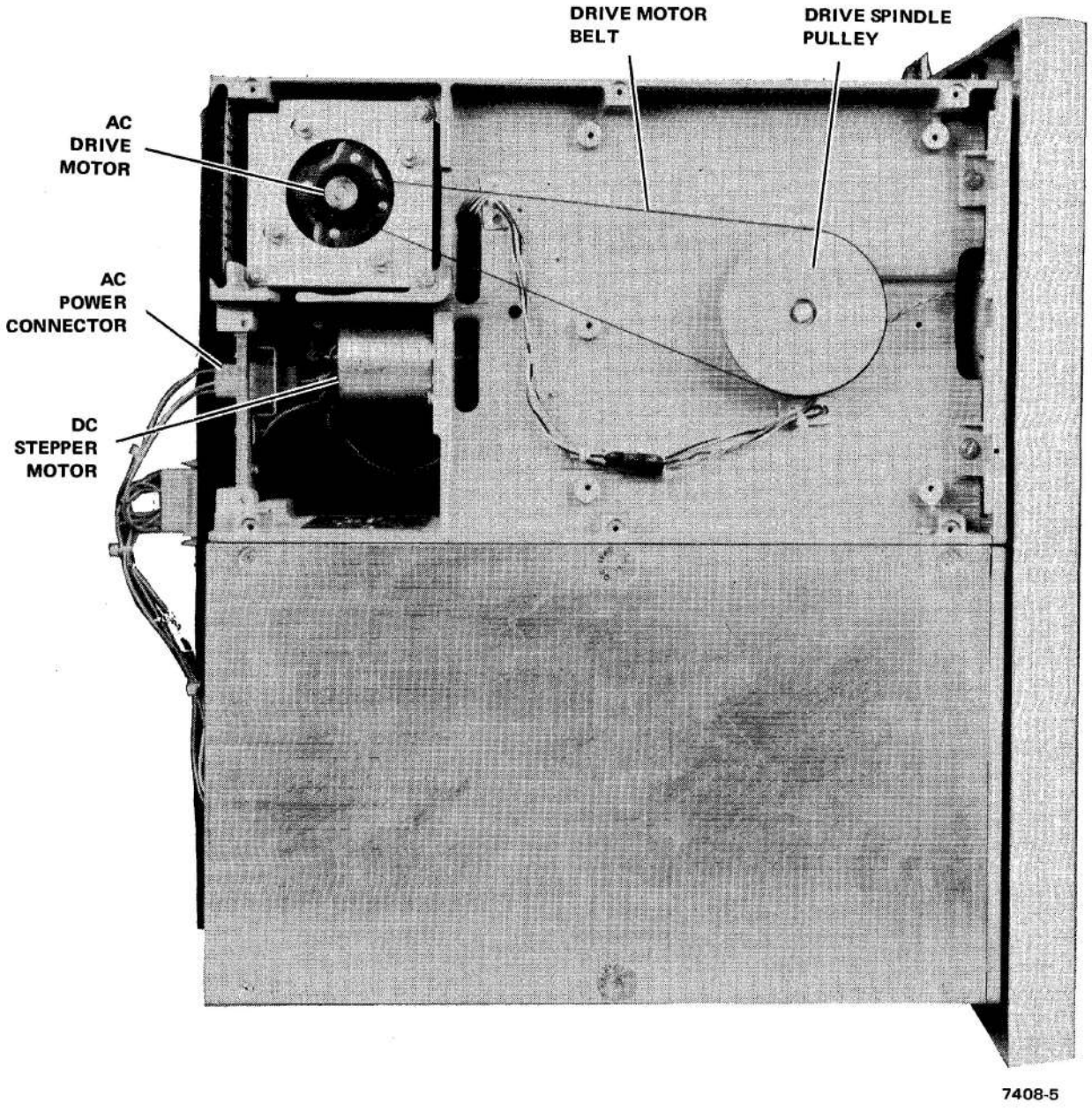


Figure 1-6 Underside View of Drive

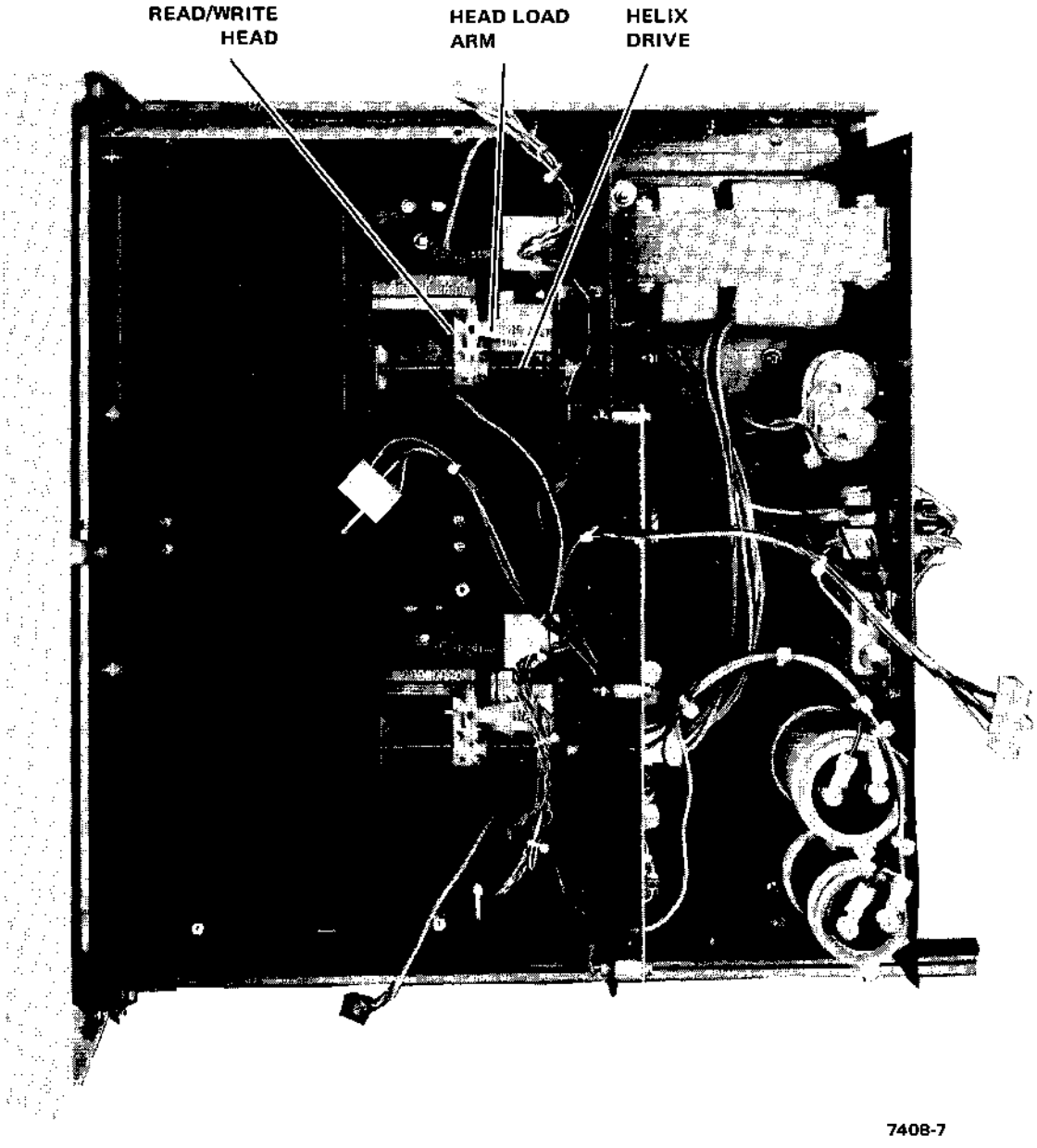


Figure 1-7 Top View of Drive

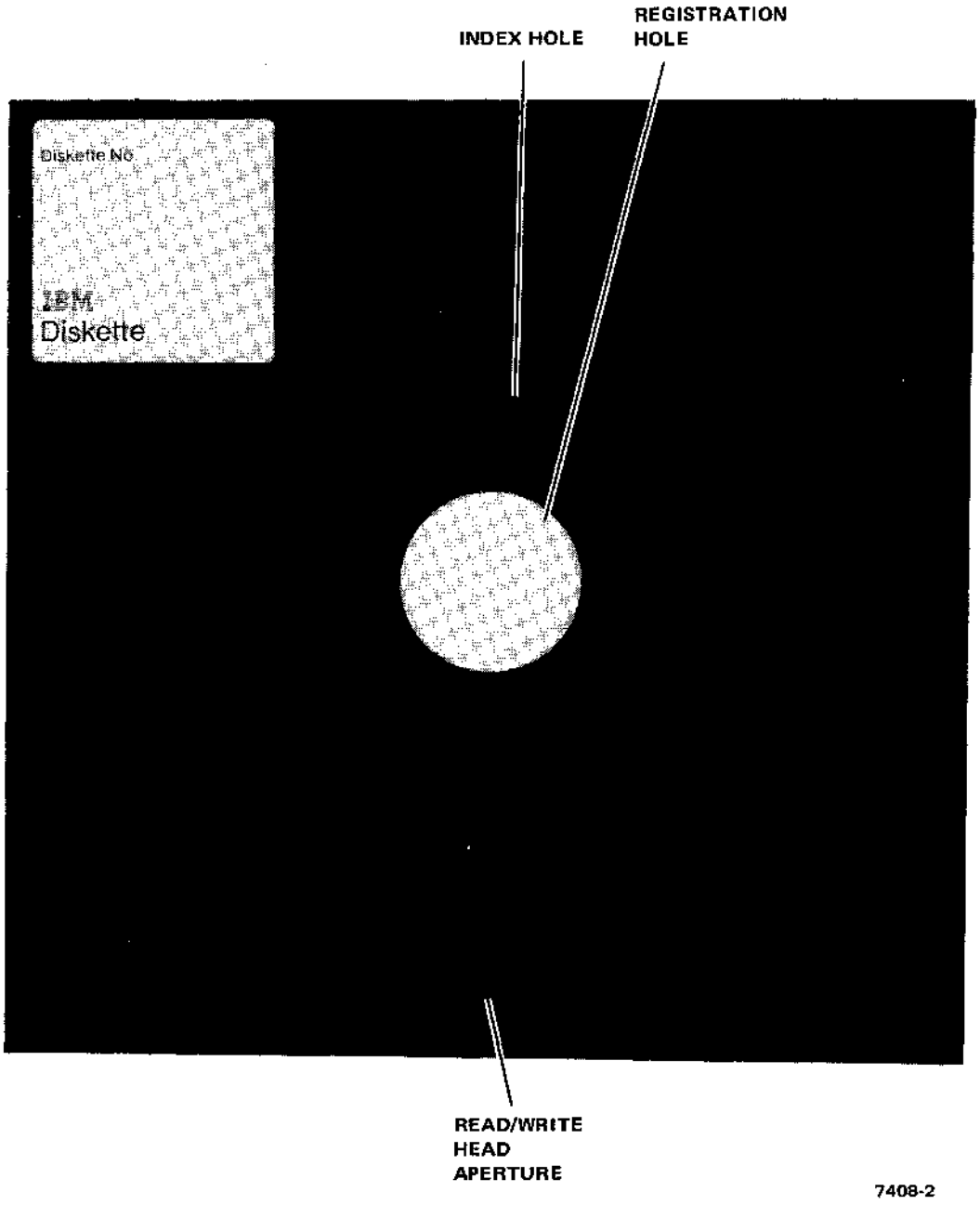


Figure 1-8 Diskette Media

1.3.2 Recording Scheme

The recording scheme used is "double frequency." In this method, data is recorded between bits of a constant clock stream. The clock stream consists of a continuous pattern of 1 flux reversal every 4 μs (Figure 1-9). A data "one" is indicated by an additional reversal between clocks (i.e., doubling the bit stream frequency; hence the name). A data "zero" is indicated by no flux reversal between clocks.

A continuous stream of ones, shown in the bottom waveform in Figure 1-9, would appear as a "2F" bit stream, and a continuous stream of zeros, shown in the top waveform in Figure 1-9, would appear as a "1F" or fundamental frequency bit stream.

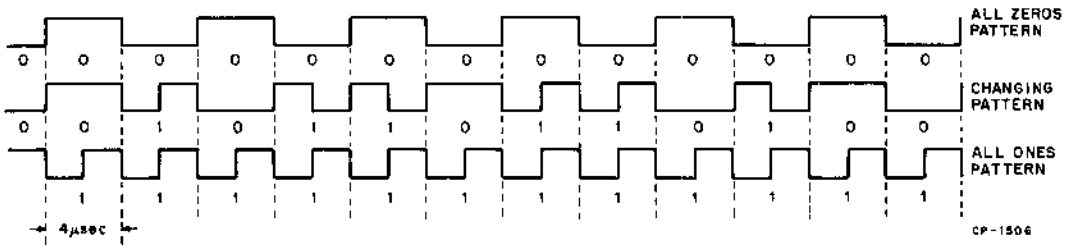


Figure 1-9 Flux Reversal Patterns

1.3.3 Logical Format

The logical format of the RX8 and RX11 Floppy Disk Systems is the same as that used in the IBM 3740.

Data is recorded on only one side of the diskette. This surface is divided into 77 concentric circles or "tracks" numbered 0–76. Each track is divided into 26 sectors numbered 1–26 (Figure 1-10). Each sector contains two major fields: the header field and the data field (Figure 1-11).

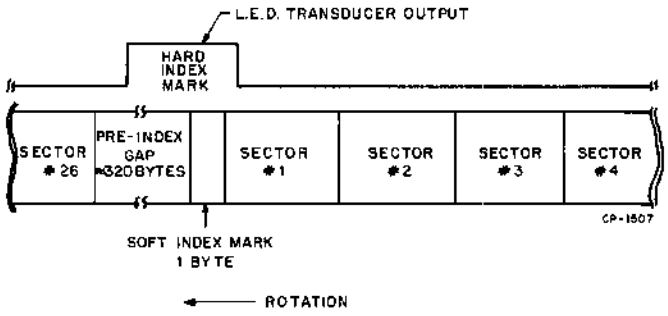


Figure 1-10 Track Format (Each Track)

1.3.3.1 Header Description – The header field is broken into seven bytes (eight bits/byte) of information and is preceded by a field of zeros for synchronization.

1. Byte No. 1: ID Address Mark – This is a unique stream of flux reversals (not a string of data bits) that is decoded by the controller to identify the beginning of the header field.

2. Byte No. 2: Track Address – This is the absolute ($0–114_8$) binary track address. Each sector contains track address information to identify its location on 1 of the 77 tracks.

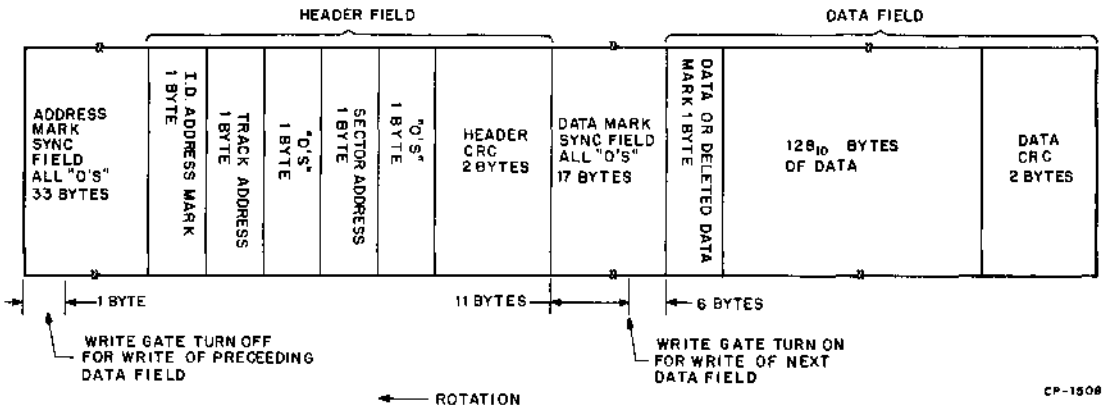


Figure 1-11 Sector Format (Each Sector)

3. Byte No. 3 – Zeros (one byte)

4. Byte No. 4: Sector Address – This is the absolute binary sector address ($1–32_8$). Each sector contains sector address information to identify its circumferential position on a track.

5. Byte No. 5 – Zeros (one byte)

6. Bytes No. 6 and 7: CRC – This is the Cyclic Redundancy Check character that is calculated for each sector from the first five header bytes using a polynomial division algorithm designed to detect the types of failures most likely to occur with "double frequency" recorded data and the floppy media. The CRC is compatible with IBM 3740 series equipment.

1.3.3.2 Data Field Description – The data field is broken into 131 bytes of information and is preceded by a field of zeros for synchronization and the header field (Figure 1-11).

1. Byte No. 1: Data or Deleted Data Address Mark – This is a unique string of flux reversals (not a string of data bits) that is decoded by the controller to identify the beginning of the data field. The deleted data mark is not used during normal operation but the RX01 can identify and write deleted data marks under program control, as required. The deleted data mark is only included in the RX8/RX11 system to be IBM compatible. One or the other data address marks precedes each data field.

2. Bytes No. 2–129 – These bytes comprise the data field used to store 128 8-bit bytes of information.

NOTE
Partial data fields are not recorded.

3. Bytes No. 130 and 131 – These bytes comprise the CRC character that is calculated for each sector from the first 129 data field bytes using the industry standard polynomial division algorithm designed to detect the types of failures most likely to occur in double frequency recording on the floppy media.

1.3.3.3 Track Usage – In the IBM 3740 system, some tracks are commonly designated for special purposes such as error information, directories, spares, or unused tracks. The RX01 is capable of recreating any system structure through the use of special systems programs, but normal operation will make use of all the available tracks as data tracks. Any special file structures must be accomplished through user software.

1.3.3.4 **CRC Capability** – Each sector has a two-byte header CRC character and a two-byte data CRC character to ensure data integrity. The CRC characters are generated by the hardware during a write operation and checked to ensure all bits were read correctly during a read operation. The CRC character is the same as that used in the IBM 3740 series of equipment. A complete description of CRC generation and checking is presented in Paragraph 5.2.3.

1.4 APPLICABLE INSTRUCTION MANUALS

This manual is designed to be used in conjunction with the RX8/RX11 Engineering Drawings. Other documents useful in operating and understanding the RX8/RX11 system are:

PDP-11 Processor Handbook*

PDP-11 Peripherals and Interfacing Handbook

PDP-8 Small Computer Handbook

PDP-8A User Manual

1.5 CONFIGURATION

Option number designations are as follows:

PDP-8 Systems

RX8-AA	Single drive system, 115 V, 60 Hz
RX8-AD	Single drive system, 50 Hz
RX8-BA	Dual drive system, 115 V/60 Hz
RX8-BD	Dual drive system, 50 Hz

PDP-11 Systems

RX11-AA	Single drive system, 115 V/60 Hz
RX11-AC	Single drive system, 50 Hz
RX11-BA	Dual drive system, 115 V/60 Hz
RX11-BD	Dual drive system, 50 Hz

NOTE

50 Hz versions are available in voltages of 105, 115, 220, 240 Vac by field pluggable conversion. See Paragraph 2.2.3.2 for complete input power modification requirements.

*Appropriate handbook for the particular processor used with the system.

1.6 SPECIFICATIONS

System Reliability

Minimum number of revolutions per track	1 million/media (head loaded)
Seek error rate	1 in 10^6 seeks
Soft read error rate	1 in 10^9 bits read
Hard read error rate	1 in 10^{12} bits read

NOTE

The above error rates *only* apply to media that is properly cared for. Seek error and soft read errors are usually attributable to random effects in the head/media interface, such as electrical noise, dirt, or dust. Both are called "soft" errors if the error is recoverable in ten additional tries or less. "Hard" errors cannot be recovered. Seek error retries should be preceded by an Initialize.

Drive Performance

Capacity	8-bit bytes	12-bit words
Per diskette	256,256 bytes	128,128 words
Per track	3,328 bytes	1,664 words
Per sector	128 bytes	64 words

Data transfer rate	
Diskette to controller buffer	4 μs/data bit (250K bps)
Buffer to CPU interface	2 μs/bit (500K bps)
CPU interface to I/O bus	18 μs/8-bit byte (>50K bytes/sec)

NOTE

PDP-8 interface can operate in 8- or 12-bit modes under software control. The transfer rate is 23 μs per 12-bit word (>40K bytes/sec).

Track-to-track move	10 ms/track maximum
Head settle time	25 ms maximum
Rotational speed	360 rpm ± 2.5%; 166 ms/rev nominal
Recording surfaces per disk	1
Tracks per disk	77 (0–76) or ($0–114_8$)
Sectors per track	26 (1–26) or ($1–32_8$)
Recording technique	Double frequency
Bit density	3200 bpi at inner track
Track density	48 tracks/in.
Average access	488 ms, computed as follows:

Seek		Settle		Rotate
(77 tks/2) X 10 ms	+	25 ms	+	(166 ms/2) = 493 ms

Environmental Characteristics

Temperature	
RX01, operating	15° to 32° C (59° to 90° F) ambient; maximum temperature gradient = 20° F/hr (11.1° C/hr)
RX01, nonoperating	-35° to +60° C (-30° to +140° F)
Media, operating	10° to 52° C (50° to 125° F)
Media, nonoperating	-35° to +52° C (-30° to +125° F)

NOTE

Media temperature must be within operating temperature range before use.

Relative humidity	
RX01, operating	25° C (77° F) maximum wet bulb 2° C (36° F) minimum dew point 20% to 80% relative humidity
RX01, nonoperating	5% to 98% relative humidity (no condensation)
Media, nonoperating	10% to 80% relative humidity
Magnetic field	Media exposed to a magnetic field strength of 50 oersteds or greater may lose data.
Interface modules	
Operating temperature	5° to 50° C (41° to 122° F)
Relative humidity	10% to 90%
Maximum wet bulb	32° C (90° F)
Minimum dew point	2° C (36° F)

Electrical

Power consumption	
RX01	3 A at 24 V (dual), 75W; 5 A at 5 V, 25 W
PDP-11 interface (M7846)	Not more than 1.5 A at 5 Vdc
PDP-8 interface (M8357)	Not more than 1.5 A at 5 Vdc
AC power input	4 A at 115 Vac 2 A at 230 Vac

CHAPTER 2
INSTALLATION AND OPERATION

2.1 PURPOSE AND ORGANIZATION

This chapter provides information on installing and operating the RX8/RX11 Floppy Disk System. This information is organized into four sections as outlined below.

1. Site Preparation – The planning required to make the installation site suitable for operation of the floppy disk system, including space, cabling, and power requirements, and fire and safety precautions.

2. Environmental Considerations – The specific environmental characteristics of the floppy disk systems, i.e., temperature, relative humidity, air conditioning and/or heat dissipation, and cleanliness.

3. Installation – The actual step-by-step process of installing the floppy disk system from unpacking through the preliminary installation checks, power conversion techniques, and acceptance testing.

4. Operation Practices – The recommended practices for using the floppy disk system, handling the media, and shipping and storing the diskettes.

2.2 SITE PREPARATION

2.2.1 Space

The RX01 is a cabinet-mountable unit that may be installed in a standard Digital Equipment Corporation cabinet. This rack-mountable version is approximately 10-1/2 in. (28 cm) high, 19 in. (48 cm) wide, and 16-1/2 in. (42 cm) deep (Figure 2-1).

Provision should be made for service clearances of approximately 22 in. (56 cm) at the front and rear of the cabinet (Figure 2-2).

2.2.2 Cabling

The standard interface cable provided with an RX8/RX11 (BC05L-15) is 15 ft (4.6 m) in length, and the positioning of the RX01 in relation to the central processor should be planned to take this into consideration. The RX01 should be placed near the control console or keyboard so that the operator will have easy access to load or unload disks. The position immediately above the CPU is preferred. The ac power cord will be about 9 ft (2.7 m) long.

2.2.3 AC Power

2.2.3.1 Power Requirements – The RX01 is designed to use either a 60 Hz or a 50 Hz power source. The 60 Hz version (RX01-A) will operate from 90 to 132 Vac, without modifications, and will use less than 4 A operating. The 50 Hz version (RX01-D) will operate within four voltage ratings and will require field verification/modification to ensure that the correct voltage option is selected. The voltage ranges of 90 to 120 Vac and 180 to 240 Vac will use less than 4 A operating. The voltage ranges of 100–132 Vac and 200–264 Vac will use less than 2 A. Both versions of the RX01 will be required to receive the input power from an ac source (e.g., 861 power control) that is controlled by the system's power switch.

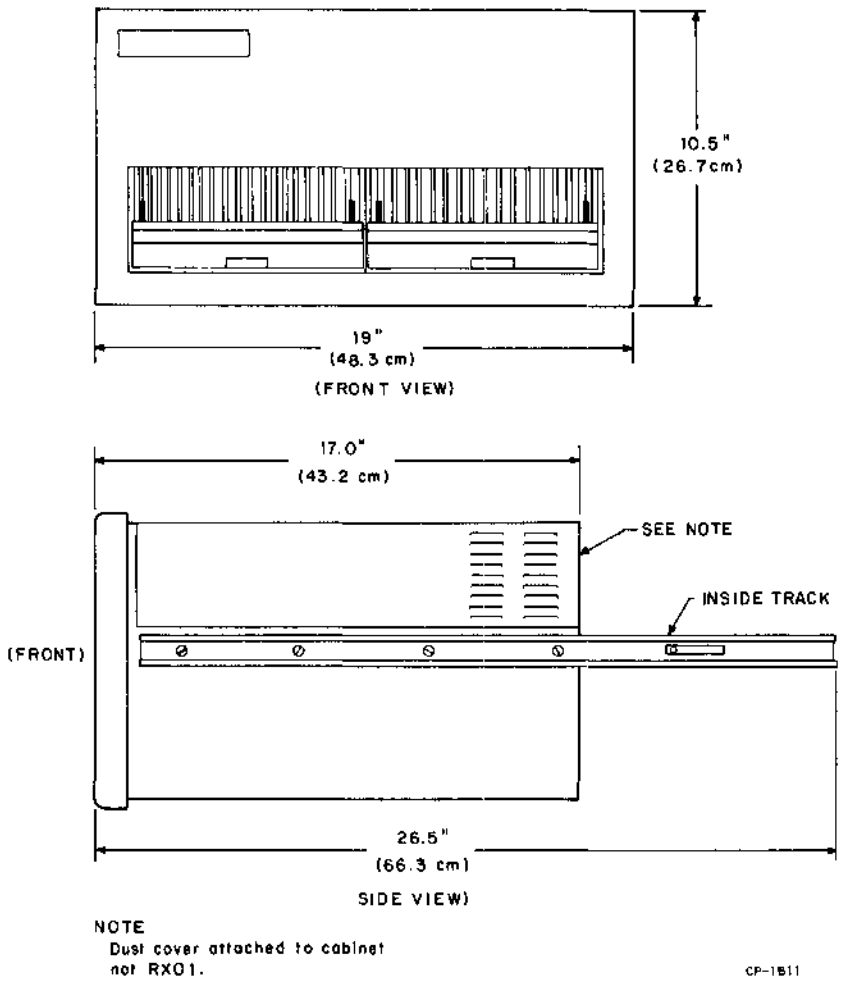


Figure 2-1 RX01

2.2.3.2 Input Power Modification Requirements – The 60 Hz version of the RX01 uses the H771A power supply and will operate on 90 to 132 Vac, without modification. To convert to operate on a 50 Hz power source in the field, the H771A supply must be replaced with an H771C or D (Figure 1-5) and the drive motor belt and drive motor pulley must be replaced (Figure 1-6). The 50 Hz version of the RX01 uses either the H771C or D power supply. The H771C operates on a 90–120 Vac or 100–132 Vac power source. The H771D operates on a 180–240 Vac or 200–264 Vac power source. To convert the H771C to the higher voltage ranges or the H771D to the lower voltage ranges, the power harness and circuit breaker must be changed. See Figure 2-3 for appropriate power harness and circuit breaker.

2.2.4 Fire and Safety Precautions

The RX8/RX11 Floppy Disk System presents no additional fire or safety hazards to an existing computer system. Wiring should be carefully checked, however, to ensure that the capacity is adequate for the added load and for any contemplated expansion.

2.3 ENVIRONMENTAL CONSIDERATIONS

2.3.1 General

The RX8/RX11 is capable of efficient operation in computer environments; however, the parameters of the operating environment must be determined by the most restrictive facets of the system, which in this case are the diskettes.

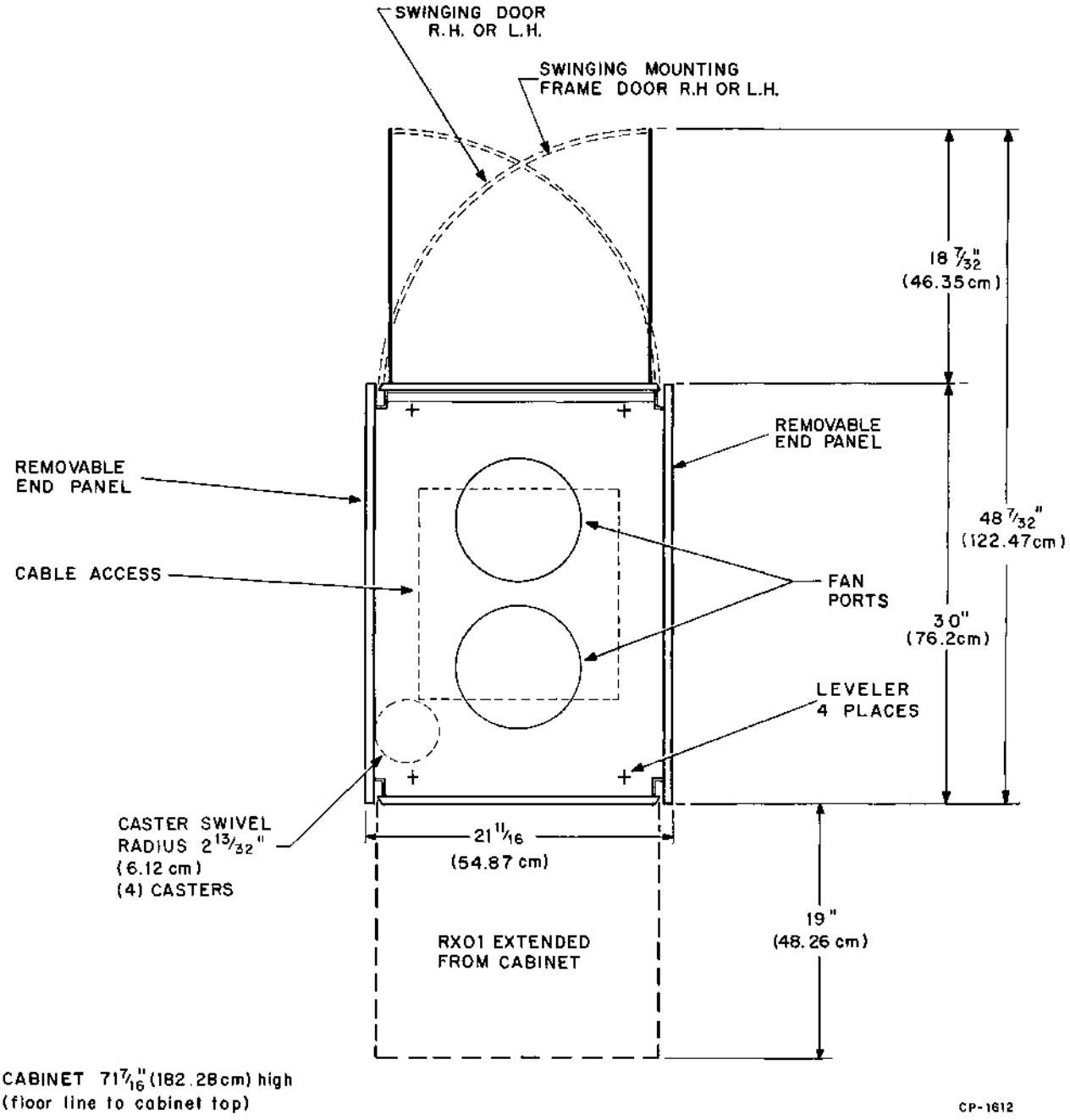


Figure 2-2 Cabinet Layout Dimensions

2.3.2 Temperature, Relative Humidity

The operating ambient temperature range of the diskette is 59° to 90° F (15° to 32° C) with a maximum temperature gradient of 20° F/hr (-6.7° C/hr).

The media nonoperating temperature range (storage) is increased to -30° to 125° F (-34.4° to 51.6° C), but care must be taken to ensure that the media has stabilized within the operating temperature range before use. This range will ensure that the media will not be operated above its absolute temperature limit of 125° F.

Humidity control is important in any system because static electricity can cause errors in any CPU with memory. The RX01 is designed to operate efficiently within a relative humidity range of 20 to 80 percent, with a maximum wet bulb temperature of 77° F (25° C) and a maximum dew point of 36° F (2° C).

2.3.3 Heat Dissipation

The heat dissipation factor for the RX01 Floppy Disk System is less than 225 Btu/hr. By adding this figure to the total heat dissipation for the other system components and then adjusting the result to compensate for such factors as the number of personnel, the heat radiation from adjoining areas, and sun exposure through windows, the approximate cooling requirements for the system can be determined. It is advisable to allow a safety margin of at least 25 percent above the maximum estimated requirements.

2.3.4 Radiated Emissions

Sources of radiation, such as FM, vehicle ignitions, and radar transmitters located close to the computer system, may affect the performance of the RX8/RX11 Floppy Disk System because of the possible adverse effects magnetic fields can have on diskettes. A magnetic field with an intensity of 50 oersteds or greater might destroy all or some of the information recorded on the diskette.

2.3.5 Cleanliness

Although cleanliness is important in all facets of a computer system, it is particularly important in the case of moving magnetic media, such as the RX01. Diskettes are not sealed units and are vulnerable to dirt. Such minute obstructions as dust specks or fingerprint smudges may cause data errors. Therefore, the RX01 should not be subjected to unusually contaminated atmospheres, especially one with abrasive airborne particles. (Refer to Paragraph 2.5.2.)

NOTE

Removable media involve use, handling, and maintenance which are beyond DEC's direct control. DEC disclaims responsibility for performance of the equipment when operated with media not meeting DEC specifications or with media not maintained in accordance with procedures approved by DEC. DEC shall not be liable for damages to the equipment or to media resulting from such operation.

2.4 INSTALLATION

2.4.1 General

The RX8/RX11 Floppy Disk System can be shipped in a cabinet as an integral part of a system or in a separate container. If the RX01 is shipped in a cabinet, the cabinet should be positioned in the final installation location before proceeding with the installation.

2.4.2 Tools

Installation of an RX8/RX11 Floppy Disk System requires no special tools or equipment. Normal hand tools are all that are necessary. However, a forklift truck or pallet handling equipment may be needed for receiving and installing a cabinet-mounted system.

2.4.3 Unpacking and Inspection

2.4.3.1 Cabinet-Mounted

1. Remove the protective covering over the cabinet.

2. Remove the restraint on the rear door latch and open the door.

3. Remove the two bolts on the cabinet's lower side rails that attach the cabinet to the pallet.

4. Raise the four levelers at the corners of the cabinet, allowing the cabinet to roll on the casters.

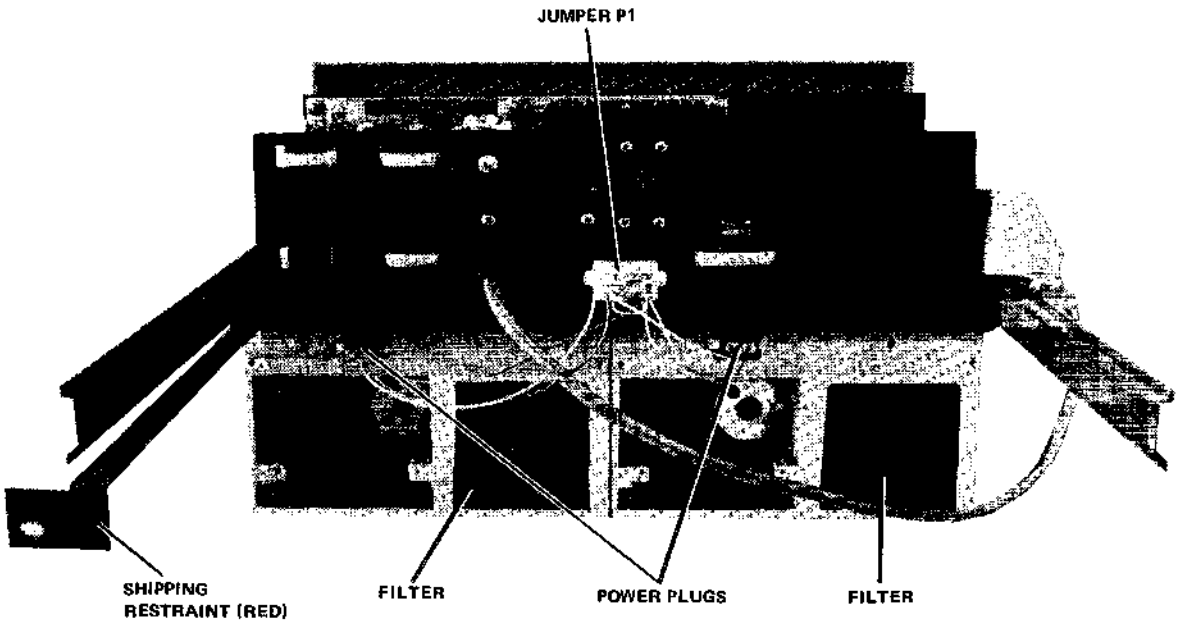
5. Carefully roll the cabinet off the pallet; if a forklift is available, it should be used to lift and move the cabinet.

6. Remove the shipping restraint from the RX01 and save it for possible reuse (Figure 2-3).

7. Slide the RX01 out on the chassis slides and visually inspect for any damage, loose screws, loose wiring, etc.

NOTE

If any shipping damage is found, the customer should be notified at this time so he can contact the carrier, and record the information on the acceptance form.



VOLTAGE (Vac)	POWER HARNESS	CIRCUIT BREAKER
90–120	70-10696-02	3.5 A, 12-12301-01
100–132	70-10696-01	3.5 A, 12-12301-01
180–240	70-10696-04	1.75 A, 12-12301-00
200–264	70-10696-03	1.75 A, 12-12301-00

Figure 2-3 RX01 Shipping Restraints

2.4.3.2 Separate Container

1. Open the carton (Figure 2-4) and remove the corrugated packing pieces.
2. Lift the RX01 out of the carton and remove the plastic shipping bag.
3. Remove the shipping fixtures from both sides of the RX01 and inspect for shipping damage.
4. Attach the inside tracks of the chassis slides provided in the carton to the RX01 (Figure 2-1).
5. Locating the proper holes in the cabinet rails (Figure 2-5), attach the outside tracks to the cabinet.
6. Place the tracks attached to the RX01 inside the extended cabinet tracks and slide the unit in until the tracks lock in the extended position.
7. Locate the RX01 cover in the cabinet above the unit and secure it to the cabinet rails (Figure 2-3).

2.4.4 Installation

1. Loosen the screws securing the upper module (M7726) and swing it up on the hinge.
2. Inspect the wiring and connectors for proper routing and ensure that they are seated correctly.
3. This step is for 50 Hz versions only. Check the power configuration to ensure that the proper power harness and the correct circuit breaker are installed (Figure 2-3).
4. Connect the BC05L-15 cable to the M7726 module and route it through the back of the RX01 (Figure 2-6) to the CPU, then connect it to the interface module (RX8E, M8357; RX11, M7846).
5. Refer to Table 2-1 for correct device code or addressing jumpers.
6. Ensure that power for the system is off.
7. Insert the interface module into the Omnibus (RX8E) or available SPC slot (RX11). (Refer to *PDP-11 Processor Handbook*, Specifications, Chapter 9.)
8. Connect the RX01 ac power cord into a switched power source.
9. Turn the power on, watching for head movement on the drive(s) during the power up, initialize phase. The head(s) should move ten tracks toward the center and back to track 0.
10. Perform the diagnostic in the sequence listed below for the number of passes (time) indicated. If any errors occur, refer to Chapter 6 for corrective action.

    RX8 or RX11 Diagnostic – 2 passes
    Data Reliability/Exerciser – 3 passes
    DECX-8 or DECX-11 – 10 minutes

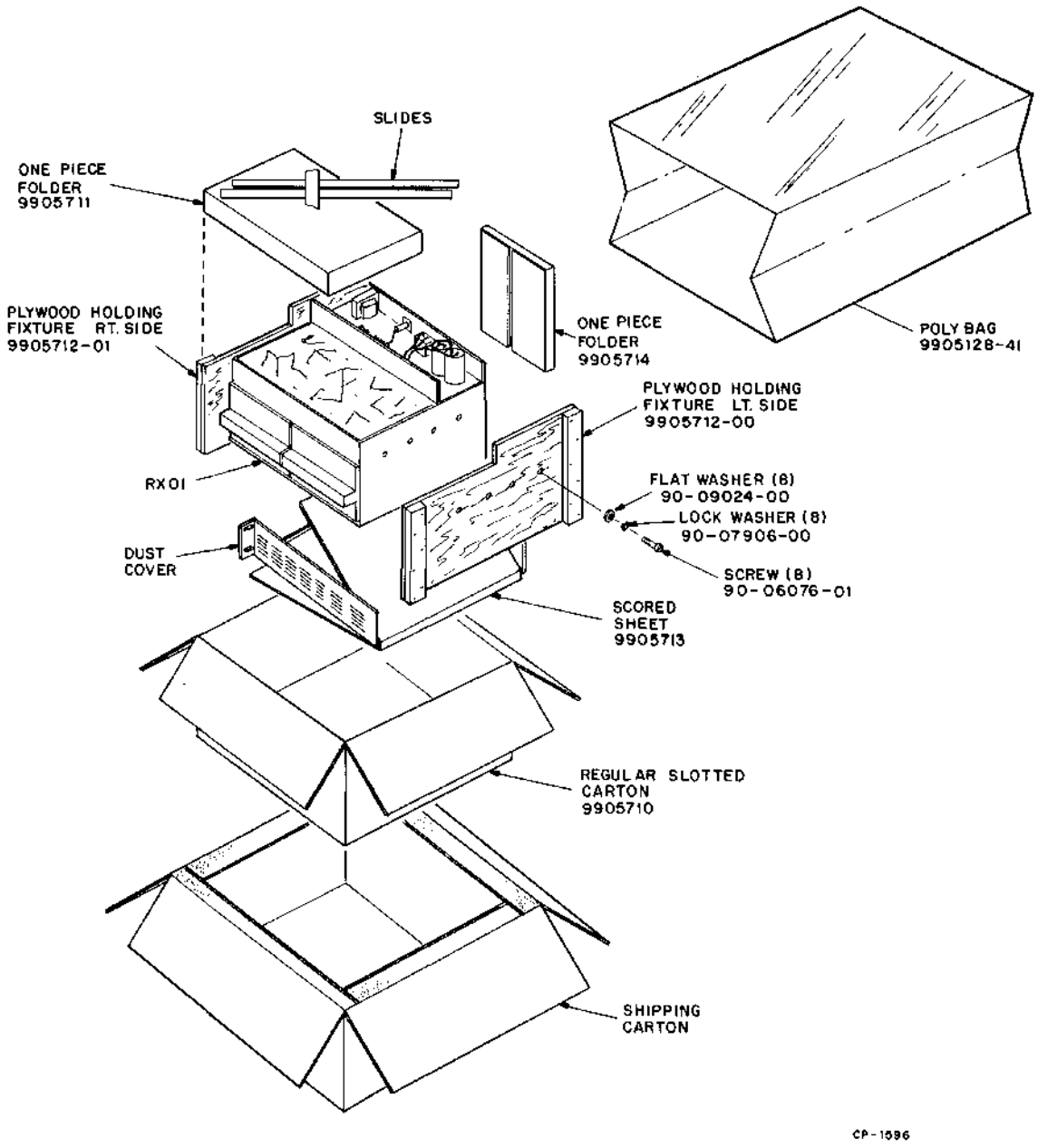


Figure 2-4 RX8/RX11 Unpacking

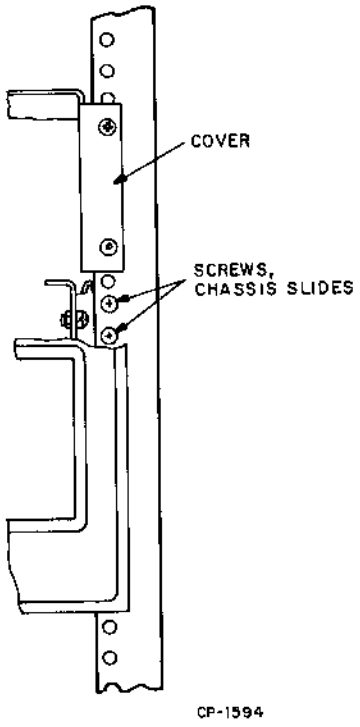


Figure 2-5 RX01 Cabinet Mounting Information

2.5 OPERATION

2.5.1 Operator Control

The simplicity of the RX01 precludes the necessity of operator controls and indicators. A convenient method of opening the unit for diskette insertion and removal is provided. On each drive is a simple pushbutton, which is compressed to allow the spring-loaded front cover to open. The diskette may be inserted or removed, as shown in Figure 2-7, with the label up. The front cover will automatically lock when the bar is pushed down.

CAUTION

The drive(s) should not be opened while they are being accessed because data may be incorrectly recorded, resulting in a CRC error when the sector is read.

2.5.2 Diskette Handling Practices and Precautions

To prolong the diskette life and prevent errors when recording or reading, reasonable care should be taken when handling the media. The following handling recommendations should be followed to prevent unnecessary loss of data or interruptions of system operation.

1. Do not write on the envelope containing the diskette. Write any information on a label prior to affixing it to the diskette.

2. Paper clips should not be used on the diskette.

3. Do not use writing instruments that leave flakes, such as lead or grease pencils, on the jacket of the media.

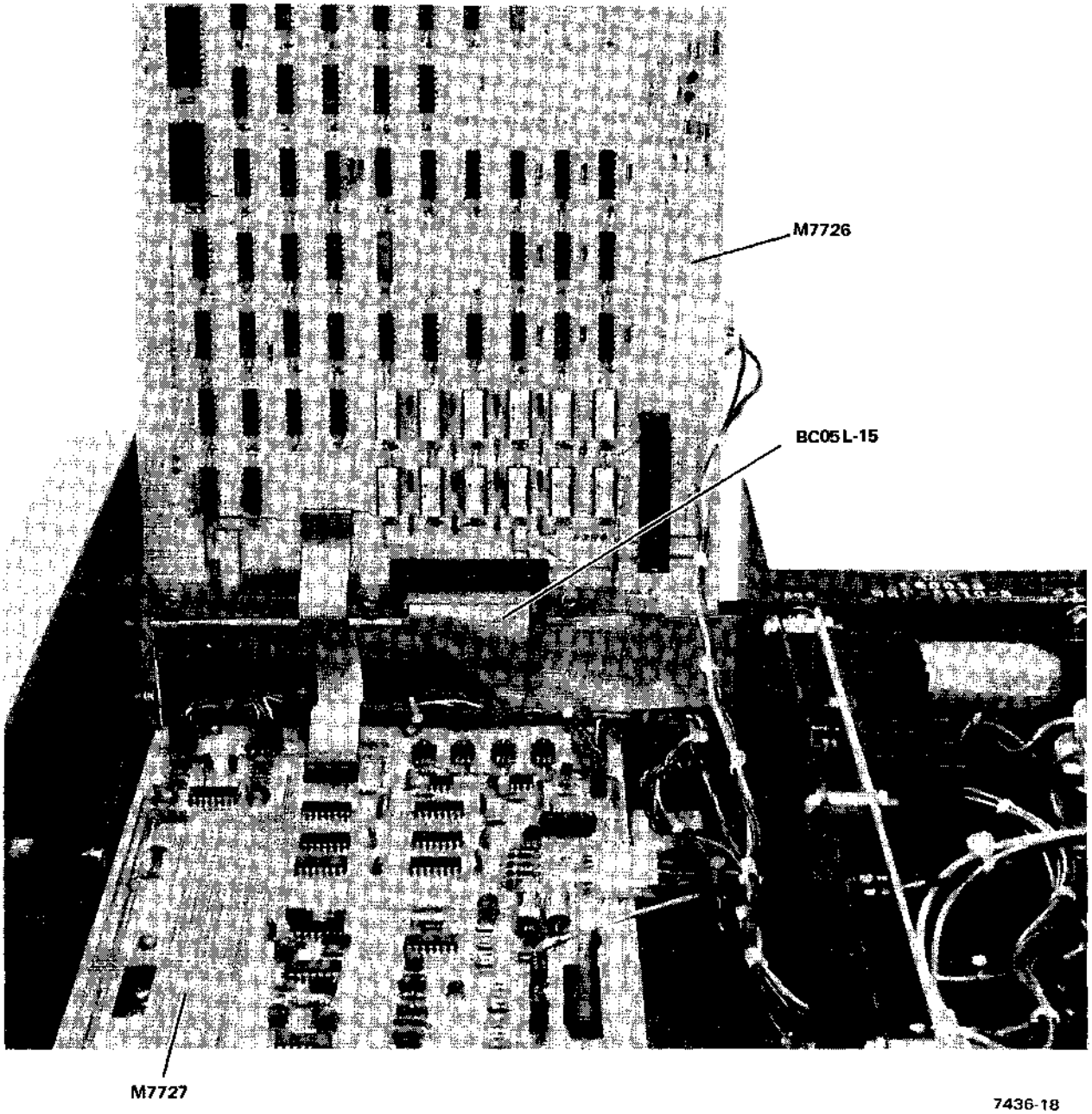


Figure 2-6 Cable Routing, BC05L-15

Table 2-1
Interface Code/Jumper Configuration

RX8E (M8357)

Device Codes

	SW1	SW2	SW3	SW4	SW5	SW6
*670X	ON	ON	ON	OFF	OFF	OFF
671X	ON	ON	OFF	OFF	OFF	ON
672X	ON	OFF	ON	OFF	ON	OFF
673X	ON	OFF	OFF	OFF	ON	ON
674X	OFF	ON	ON	ON	OFF	OFF
675X	OFF	ON	OFF	ON	OFF	ON
676X	OFF	OFF	ON	ON	ON	OFF
677X	OFF	OFF	OFF	ON	ON	ON

RX11 (M7846)

BR Priority

BR7 – 54-08782
BR6 – 54-08780
*BR5 – 54-08778
BR4 – 57-08776

**Unibus Address 17717X*

A12/W18 – Removed
A11/W17 – Removed
A10/W16 – Removed
A9/W15 – Removed
A8/W14 – Installed
A7/W13 – Installed
A6/W12 – Removed
A5/W11 – Removed
A4/W10 – Removed
A3/W9 – Removed

**Vector Address (264_8)*

V2/W1 – Installed
V3/W2 – Removed
V4/W3 – Installed
V5/W4 – Installed
V6/W5 – Removed
V7/W6 – Installed
V8/W7 – Removed

*Standard

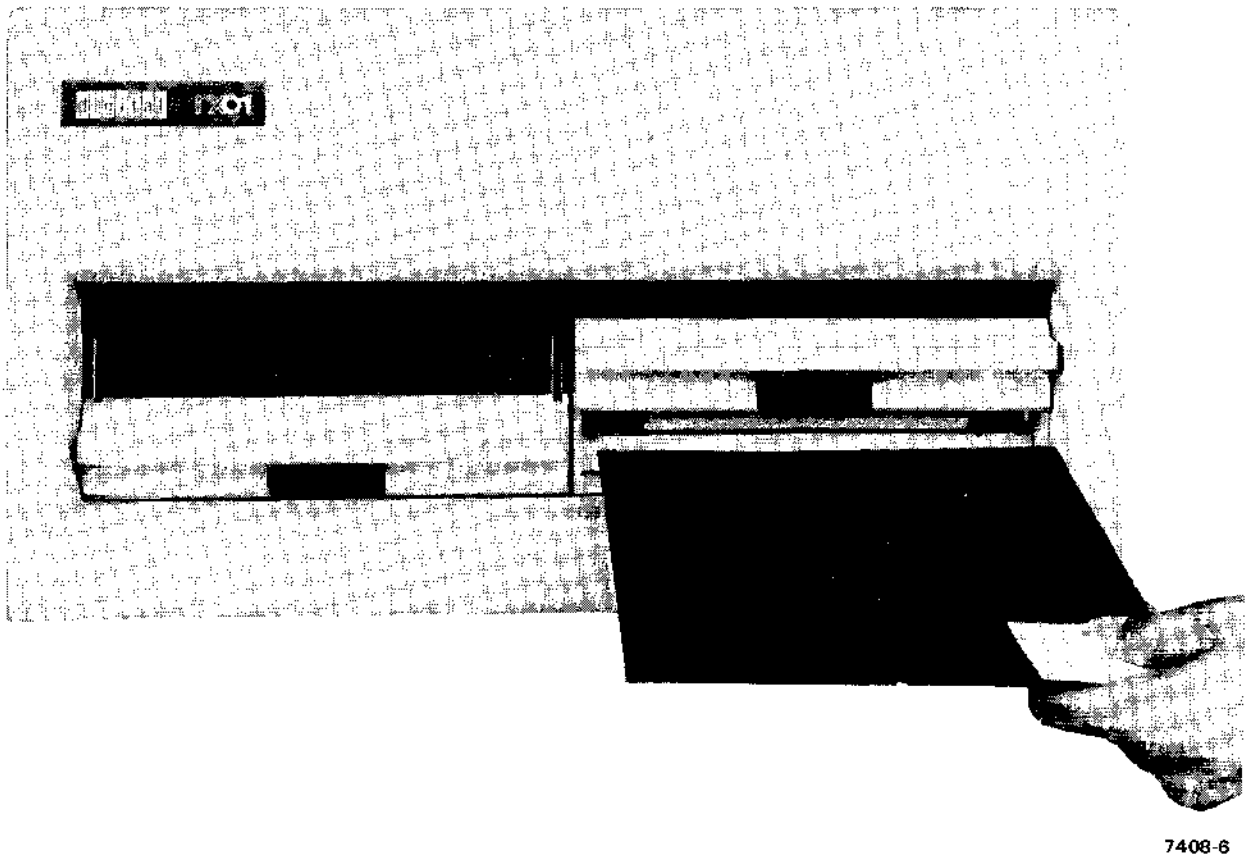


Figure 2-7 Flexible Diskette Insertion

4. Do not touch the disk surface exposed in the diskette slot or index hole.

5. Do not clean the disk in any manner.

6. Keep the diskette away from magnets or tools that may have become magnetized. Any disk exposed to a magnetic field may lose information.

7. Do not expose the diskette to a heat source or sunlight.

8. Always return the diskette to the envelope supplied with it to protect the disk from dust and dirt. Diskettes not being used should be stored in the file box if possible.

9. When the diskette is in use, protect the empty envelope from liquids, dust, and metallic materials.

10. Do not place heavy items on the diskette.

11. Do not store diskettes on top of computer cabinets or in places where dirt can be blown by fans into the diskette interior.

12. If a diskette has been exposed to temperatures outside of the operating range, allow 5 minutes for thermal stabilization before use. The diskette should be removed from its packaging during this time.

2.5.3 Diskette Storage

2.5.3.1 Short Term (Available for Immediate Use)

1. Store diskettes in their envelopes.

2. Store horizontally, in piles of ten or less. If vertical storage is necessary, the diskettes should be supported so that they do not lean or sag, but should not be subjected to compressive forces. Permanent deformation may result from improper storage.

3. Store in an environment similar to that of the operating system; at a minimum, store within the operating environment range.

2.5.3.2 Long Term – When diskettes do not need to be available for immediate use, they should be stored in their original shipping containers within the nonoperating range of the media.

2.5.4 Shipping Diskettes

Data recorded on disks may be degraded by exposure to any sort of small magnet brought into close contact with the disk surface. If diskettes are to be shipped in the cargo hold of an aircraft, take precautions against possible exposure to magnetic sources. Because physical separation from the magnetic source is the best protection against accidental erasure of a diskette, diskettes should be packed at least 3 in. within the outer box. This separation should be adequate to protect against any magnetic sources likely to be encountered during transportation, making it generally unnecessary to ship diskettes in specially shielded boxes.

When shipping, be sure to label the package:

DO NOT EXPOSE TO PROLONGED HEAT OR SUNLIGHT.

When received, the carton should be examined for damage. Deformation of the carton should alert the receiver to possible damage of the diskette. The carton should be retained, if it is intact, for storage of the diskette or for future shipping.

CHAPTER 3
RX11 INTERFACE PROGRAMMING INFORMATION

This chapter describes device registers, register and vector address assignments, programming specifications, and programming examples for the RX11 interface.

All software control of the RX11 is performed by means of two device registers: the RX11 Command and Status register (RXCS) and a multipurpose RX11 Data Buffer register (RXDB). These registers have been assigned bus addresses and can be read or loaded, with certain exceptions, using any instruction referring to their addresses.

The RX01, which includes the mechanical drive(s), read/write electronics, and μCPU controller, contains all the control circuitry required for implied seeks, automatic head position verification, and calculation and verification of the CRC; it has a buffer large enough to hold one full sector of diskette data (128 8-bit bytes). Information is serially passed between the interface and the RX01.

A typical diskette write sequence, which is initiated by a user program, would occur in two steps:

1. Fill Buffer – A command to fill the buffer is moved into the RXCS. The Go bit (Paragraph 3.2.1) must be set. The program tests for Transfer Request (TR). When TR is detected, the program moves the first of 128 bytes of data to the RXDB. TR goes false while the byte is moved into the RX01. The program retests TR and moves another byte of data when TR is true. When the RX01 sector buffer is full, the Done bit will set, and an interrupt will occur if the program has enabled interrupts.

2. Write Sector – A command to write the contents of the buffer onto the disk is issued to the RXCS. Again the Go bit must be set. The program tests TR, and when TR is true, the program moves the desired sector address to the RXDB. TR goes false while the RX01 handles the sector address. The program again waits for TR and moves the desired track address to the RXDB, and again TR is negated. The RX01 locates the desired track and sector, verifies its location, and writes the contents of the sector buffer onto the diskette. When this is done, an interrupt will occur if the program has enabled interrupts.

A typical diskette read occurs in just the reverse way: first locating and reading a sector into the buffer (Read Sector) and then unloading the buffer into core (Empty Buffer). In either case, the content of the buffer is not valid if Power Fail or Initialize follows a Fill Buffer or Read Sector function.

3.1 REGISTER AND VECTOR ADDRESSES

The RXCS register is normally assigned Unibus address 177170, and the RXDB register is assigned Unibus address 177172. The normal BR priority level is 5, but it can be changed by insertion of a different priority plug located on the interface module. The vector address is 264.

3.2 REGISTER DESCRIPTION

3.2.1 RXCS – Command and Status (177170)

Loading this register while the RX01 is not busy and with bit 0 = 1 will initiate a function as described below and indicated in Figure 3-1. Bits 0–4 write-only bits.

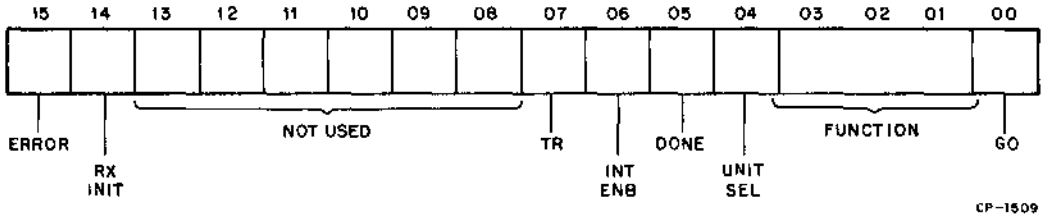


Figure 3-1 RXCS Format (RX11)

Bit No.	Description
0	Go – Initiates a command to RX01. This is a write-only bit.
1–3	Function Select – These bits code one of the eight possible functions described in Paragraph 3.3 and listed below. These are write-only bits.

Code	Function
000	Fill Buffer
001	Empty Buffer
010	Write Sector
011	Read Sector
100	Not used
101	Read Status
110	Write Deleted Data Sector
111	Read Error Register

4	Unit select – This bit selects one of the two possible disks for execution of the desired function. This is a write-only bit. Unit 0 is physically the left-hand unit in the rack.
5	Done – This bit indicates the completion of a function. Done will generate an interrupt when asserted if Interrupt Enable (RXCS bit 6) is set. This is a read-only bit.
6	Interrupt Enable – This bit is set by the program to enable an interrupt when the RX01 has completed an operation (Done). The condition of this bit is normally determined at the time a function is initiated. This bit is cleared by Initialize and is a read/write bit.
7	Transfer Request – This bit signifies that the RX11 needs data or has data available. This is a read-only bit.
8–13	Unused

Bit No.	Description
14	RX11 Initialize – This bit is set by the program to initialize the RX11 without initializing all of the devices on the Unibus. This is a write-only bit.

CAUTION

Loading the lower byte of the RXCS will also load the upper byte of the RXCS.

Upon setting this bit in the RXCS, the RX11 will negate Done and move the head position mechanism of drive 1 (if two are available) to track 0. Upon completion of a successful Initialize, the RX01 will zero the Error and Status register, set Initialize Done, and set RXES bit 7 (DRV RDY) if unit 0 is ready. It will also read sector 1 of track 1 on drive 0.

15	Error – This bit is set by the RX01 to indicate that an error has occurred during an attempt to execute a command. This read-only bit is cleared by the initiation of a new command or an Initialize (Paragraph 3.6).

3.2.2 RXDB – Data Buffer Register (177172)

This register serves as a general purpose data path between the RX01 and the interface. It may represent one of four RX01 registers according to the protocol of the function in progress (Paragraph 3.3).

This register is read/write if the RX01 is not in the process of executing a command; that is, it may be manipulated without affecting the RX01 subsystem. If the RX01 is actively executing a command, this register will only accept data if RXCS bit 7 (TR) is set. In addition, valid data can only be read when TR is set.

CAUTION

Violation of protocol in manipulation of this register may cause permanent data loss.

3.2.2.1 RXTA – RX Track Address (Figure 3-2) – This register is loaded to indicate on which of the 115_8 tracks a given function is to operate. It can be addressed only under the protocol of the function in progress (Paragraph 3.3). Bits 8 through 15 are unused and are ignored by the control.

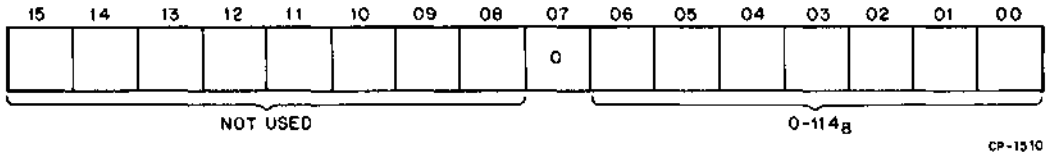


Figure 3-2 RXTA Format (RX11)

3.2.2.2 RXSA – RX Sector Address (Figure 3-3) – This register is loaded to indicate on which of the 32_8 sectors a given function is to operate. It can be addressed only under the protocol of the function in progress (Paragraph 3.3). Bits 8 through 15 are unused and are ignored by the control.

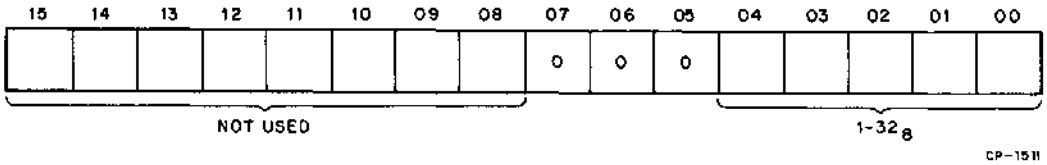


Figure 3-3 RXSA Format (RX11)

3.2.2.3 **RXDB – RX Data Buffer** (Figure 3-4) – All information transferred to and from the floppy media passes through this register and is addressable only under the protocol of the function in progress (Paragraph 3.3).

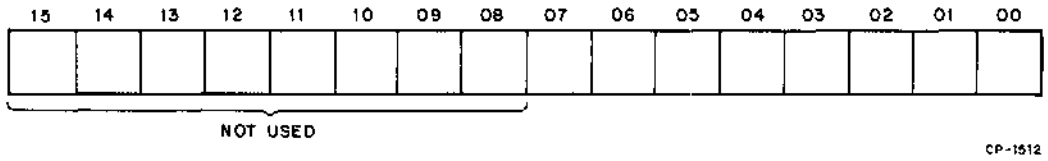


Figure 3-4 RXDB Format (RX11)

3.2.2.4 **RXES – RX Error and Status** (Figure 3-5) – This register contains the current error and status conditions of the drive selected by bit 4 (Unit Select) of the RXCS. This read-only register can be addressed only under the protocol of the function in progress (Paragraph 3.3). The RXES is located in the RXDB upon completion of a function.

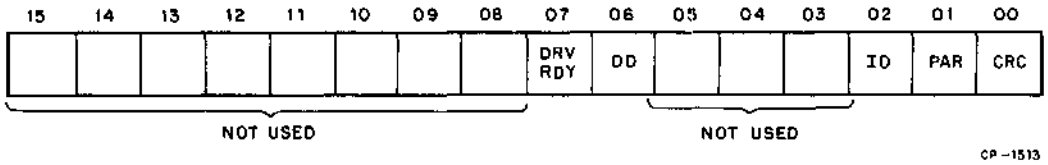


Figure 3-5 RXES Format (RX11)

RXES bit assignments are:

Bit No.	Description
0	CRC Error – A cyclic redundancy check error was detected as information was retrieved from a data field of the diskette. The RXES is moved to the RXDB, and Error and Done are asserted.
1	Parity Error – A parity error was detected on command or address information being transferred to the RX01 from the Unibus interface. A parity error indication means that there is a problem in the interface cable between the RX01 and the interface. Upon detection of a parity error, the current function is terminated; the RXES is moved to the RXDB, and Error and Done are asserted.
2	Initialize Done – This bit is asserted in the RXES to indicate completion of the Initialize routine which can be caused by RX01 power failure, system power failure, or programmable or Unibus Initialize.
3–5	Unused
6	Deleted Data Detected – During data recovery, the identification mark preceding the data field was decoded as a deleted data mark (Paragraph 1.3.3).

Bit No.	Description
7	Drive Ready – This bit is asserted if the unit currently selected exists, is properly supplied with power, has a diskette installed correctly, has its door closed, and has a diskette up to speed.

NOTE 1

The Drive Ready bit is only valid when retrieved via a Read Status function or at completion of Initialize when it indicates status of drive 0.

NOTE 2

If the Error bit was set in the RXCS but Error bits are not set in the RXES, then specific error conditions can be accessed via a Read Error Register function (Paragraph 3.3.7).

3.3 FUNCTION CODES

Following the strict protocol of the individual function, data storage and recovery on the RX11 occur with careful manipulation of the RXCS and RXDB registers. The penalty for violation of protocol can be permanent data loss.

A summary of the function codes is presented below:

000	Fill Buffer
001	Empty Buffer
010	Write Sector
011	Read Sector
100	Not used
101	Read Status
110	Write Deleted Data Sector
111	Read Error Register

The following paragraphs describe in detail the programming protocol associated with each function encoded and written into RXCS bits 1–3 if Done is set.

3.3.1 Fill Buffer (000)

This function is used to fill the RX01 buffer with 128 8-bit bytes of data from the host processor. Fill Buffer is a complete function in itself; the function ends when the buffer has been filled. The contents of the buffer can be written onto the diskette by means of a subsequent Write Sector function, or the contents can be returned to the host processor by an Empty Buffer function.

RXCS bit 4 (Unit Select) does not affect this function, since no diskette drive is involved. When the command has been loaded, RXCS bit 5 (Done) is negated. When the TR bit is asserted, the first byte of the data may be loaded into the data buffer. The same TR cycle will occur as each byte of data is loaded. The RX01 counts the bytes transferred; it will not accept less than 128 bytes and will ignore those in excess. Any read of the RXDB during the cycle of 128 transfers is ignored by the RX11.

3.3.2 Empty Buffer (001)

This function is used to empty the internal buffer of the 128 data bytes loaded from a previous Read Sector or Fill Buffer command. This function will ignore RXCS bit 4 (Unit Select) and negate Done.

When TR sets, the program may unload the first of 128 data bytes from the RXDB. Then the RX11 again negates TR. When TR resets, the second byte of data may be unloaded from the RXDB, which again negates TR. Alternate checks on TR and data transfers from the RXDB continue until 128 bytes of data have been moved from the RXDB. Done sets, ending the operation and initiating an interrupt if RXCS bit 6 (Interrupt Enable) is set.

NOTE

The Empty Buffer function does *not* destroy the contents of the sector buffer.

3.3.3 Write Sector (010)

This function is used to locate a desired track and sector and write the sector with the contents of the internal sector buffer. The initiation of this function clears bits 0, 1, and 6 of RXES (CRC Error, Parity Error, and Deleted Data Detected) and negates Done.

When TR is asserted, the program must move the desired sector address into the RXDB, which will negate TR. When TR is again asserted, the program must load the desired track address into the RXDB, which will negate TR. If the desired track is not found, the RX11 will abort the operation, move the contents of the RXES to the RXDB, set RXCS bit 15 (Error), assert Done, and initiate an interrupt if RXCS bit 6 (Interrupt Enable) is set.

TR will remain negated while the RX01 attempts to locate the desired sector. If the RX01 is unable to locate the desired sector within two diskette revolutions, the RX11 will abort the operation, move the contents of the RXES to the RXDB, set RXCS bit 15 (Error), assert Done, and initiate an interrupt if RXCS bit 6 (Interrupt Enable) is set.

If the desired sector is successfully located, the RX11 will write the 128 bytes stored in the internal buffer followed by a 16-bit CRC character that is automatically calculated by the RX01. The RX11 ends the function by asserting Done and initiating an interrupt if RXCS bit 6 (Interrupt Enable) is set.

NOTE 1

The contents of the sector buffer are not valid data after a power loss has been detected by the RX01. The Write Sector function, however, will be accepted as a valid function, and the random contents of the buffer will be written, followed by a valid CRC.

NOTE 2

The Write Sector function does *not* destroy the contents of the sector buffer.

3.3.4 Read Sector (011)

This function is used to locate a desired track and sector and transfer the contents of the data field to the μCPU controller sector buffer. The initiation of this function clears bits 0, 1, and 6 of RXES (CRC Error, Parity Error, Deleted Data Detected) and negates Done.

When TR is asserted, the program must load the desired sector address into the RXDB, which will negate TR. When TR is again asserted, the program must load the desired track address into the RXDB, which will negate TR.

If the desired track is not found, the RX11 will abort the operation, move the contents of the RXES to the RXDB, set RXCS bit 15 (Error), assert Done, and initiate an interrupt if RXCS bit 6 (Interrupt Enable) is set.

TR and Done will remain negated while the RX01 attempts to locate the desired track and sector. If the RX01 is unable to locate the desired sector within two diskette revolutions after locating the presumably correct track, the RX11 will abort the operation, move the contents of the RXES to the RXDB, set RXCS bit 15 (Error), assert Done, and initiate an interrupt if RXCS bit 6 (Interrupt Enable) is set.

If the desired sector is successfully located, the control will attempt to locate a standard data address mark or a deleted data address mark. If either mark is properly located, the control will read data from the sector into the sector buffer.

If the deleted data address mark was detected, the control will assert RXES bit 6 (DD). As data enters the sector buffer, a CRC is computed, based on the data field and CRC bytes previously recorded. A non-zero residue indicates that a read error has occurred. The control sets RXES bit 0 (CRC Error) and RXCS bit 15 (Error). The RX11 ends the operation by moving the contents of the RXES to the RXDB, sets Done, and initiates an interrupt if RXCS bit 6 (Interrupt Enable) is set.

3.3.5 Read Status (101)

The RX11 will negate RXCS bit 5 (Done) and begin to assemble the current contents of the RXES into the RXDB. RXES bit 7 (Drive Ready) will reflect the status of the drive selected by RXCS bit 4 (Unit Select) at the time the function was given. All other RXES bits will reflect the conditions created by the last command. RXES may be sampled when RXCS bit 5 (Done) is again asserted. An interrupt will occur if RXCS bit 6 (Interrupt Enable) is set. RXES bits are defined in Paragraph 3.2.2.

NOTE

The average time for this function is 250 ms. Excessive use of this function will result in substantially reduced throughput.

3.3.6 Write Sector with Deleted Data (110)

This operation is identical to function 010 (Write Sector) with the exception that a deleted data address mark precedes the data field instead of a standard data address mark (Paragraph 1.3.3.2).

3.3.7 Read Error Register Function (111)

The Read Error Register function can be used to retrieve explicit error information provided by the μCPU controller upon detection of the general error bit. The function is initiated, and bits 0–6 of the RXES are cleared. Out is asserted and Done is negated. The controller then generates the appropriate number of shift pulses to transfer the specific error code to the Interface register and completes the function by asserting Done. The Interface register can now be read and the error code interrogated to determine the type of failure that occurred (Paragraph 3.6).

NOTE

Care should be exercised in use of this function since, under certain conditions, erroneous error information may result (Paragraph 3.5).

3.3.8 Power Fail

There is no actual function code associated with Power Fail. When the RX01 senses a loss of power, it will unload the head and abort all controller action. All status signals are invalid while power is low.

When the RX01 senses the return of power, it will remove Done and begin a sequence to:

1. Move drive 1 head position mechanism to track 0.

2. Clear any active error bits.

3. Read sector 1 of track 1 of drive 0 into the sector buffer.

4. Set RXES bit 02 (Initialize Done) (Paragraph 3.2.2.4) after which Done is again asserted.

5. Set Drive Ready of the RXES according to the status of drive 0.

There is no guarantee that information being written at the time of a power failure will be retrievable. However, all other information on the diskette will remain unaltered.

A method of aborting a function is through the use of RXCS bit 14 (RX11 Initialize). Another method is through the use of the system Initialize signal that is generated by the PDP-11 RESET instruction, the console START key, or system power failure.

3.4 PROGRAMMING EXAMPLES

3.4.1 Read Data/Write Data

Figure 3-6 presents a program for implementing a Write, Write Deleted Data, or a Read function, depending on the function code that is used. The first instructions set up the error retry counters, PTRY, CTRY, and STRY. The instruction RETRY moves the command word for a Write, Write Deleted Data, or Read into the RXCS.

The set of three instructions beginning at the label 1$ moves the sector address to the RX11 after Transfer Request (TR), which is bit 7, has been set. The three instructions beginning at the label 2$ move the track address to the RX11 after TR has been set. The group of instructions beginning at the label 3$ looks for the Done flag to set and checks for errors.

An error condition, indicated by bit 15 setting, is checked beginning at ERFLAG. If bit 0 is set, a CRC error has occurred, and a branch is made to CRCER. If bit 1 is set, a parity error has occurred, and a branch is made to PARER. If neither of the above bits is set, a seek error is assumed to have occurred and a branch is made to SEEKER, where the system is initialized. In the case of a Write function, the sector buffer is refilled by a JMP to FILLBUF. In the case of a Read function, a JMP is made to EMPBUFF.

In each of the PAR, CRC, and SEEK routines, the command sequence is retried ten times by decrementing the respective retry counter. If an error persists after ten tries, it is a hard error. The retry counters can be set up to retry as many times as desired.

NOTE

A Fill Buffer function is performed before a Write function, and an Empty Buffer function is performed after a Read function.

3.4.2 Empty Buffer Function

Figure 3-7 shows a program for implementing an Empty Buffer function. The first instruction sets the number of error retries to ten. The address of the memory buffer is placed in register R0, and the Empty Buffer command is placed in the RXCS. Existence of a parity error is checked starting at instruction 3$. If a parity error is detected, the Empty Buffer command is loaded again. If an error persists for ten retries, the error is considered hard.

```
                                .ABS
                                ;PROGRAMMING EXAMPLES FOR THE RX11/RX01 FLEXIBLE DISKETTE
                                ;
                                ;THE FOLLOWING IS THE RX11 STANDARD DEVICE ADDRESS AND VECTOR ADDRESS
                                ;
        177170                  RXCS=177170                     ; COMMAND STATUS REGISTER
        177172                  RXDB=177172                     ; DATA BUFFER REGISTER
        177172                  RXSA=177172                     ; SECTOR ADDRESS REGISTER
        177172                  RXTA=177172                     ; TRACK ADDRESS REGISTER
        177172                  RXES=177172                     ; ERROR STATUS REGISTER
                                ;
                                ;THE FOLLOWING IS A PROGRAMMING EXAMPLE OF THE PROTOCOL REQUIRED
                                ;TO WRITE, WRITE DELETED DATA, OR READ AT SECTOR "S" (THE CONTENTS OF PROGRAM
                                ;LOCATION SECTOR) OF TRACK "T" (THE CONTENTS OF PROGRAM LOCATION TRACK)
                                ;
000000  012767  177770  000328  START:  MOV #-10, PTRY           ; PARITY RETRY COUNTER
000006  012767  177770  000314          MOV #-10, CTRY           ; CRC RETRY COUNTER
000014  012767  177770  000310          MOV #-10, STRY           ; SEEK RETRY COUNTER
                                ;
                                ;WRITE, WRITE DELETED DATA, OR READ
                                ;
                                ; BITS 4 THRU 1 OF PROGRAM LOCATION COMMAND CONTAIN THE FUNCTION
                                ;
                                ; BIT 4 = 1 MEANS UNIT 1 ( = 0 MEANS UNIT 0)
                                ;
                                ; BITS 3 THRU 1 IS THE COMMAND ( 4 = WRITE, 14 = WRITE DELETED DATA, 6 = READ)
                                ;
000022  016767  000306  177140  RETRY:  MOV COMMAND, RXCS         ; UNIT + (WRITE, WRITE DELETED DATA, OR READ)
                                ;
                                ;WAIT FOR THE TRANSFER REQUEST FLAG THEN TRANSFER THE SECTOR ADDRESS
                                ;
000030  105767  177134          1$:     TSTB RXCS               ; TEST FOR THE TRANSFER REQUEST FLAG
000034  001775                          BEQ 1$                  ; BEQ UNTIL THE TRANSFER REQUEST FLAG SETS
000036  116767  000274  177126          MOVB SECTOR, RXSA       ; LOAD SECTOR ADDRESS
                                ;
                                ;WAIT FOR THE TRANSFER REQUEST FLAG THEN TRANSFER THE TRACK ADDRESS
                                ;
000044  105767  177120          2$:     TSTB RXCS               ; TEST FOR THE TRANSFER REQUEST FLAG
000050  001775                          BEQ 2$                  ; BEQ UNTIL THE TRANSFER REQUEST FLAG SETS
000052  116767  000262  177112          MOVB TRACK, RXTA        ; LOAD TRACK ADDRESS
                                ;
                                ;THE SECTOR AND TRACK ADDRESSES HAVE BEEN TRANSFERRED TO THE RX01
                                ;
                                ;WAIT FOR THE DONE FLAG AND CHECK FOR ANY ERRORS
                                ;
                                ;IF THE FUNCTION HAS COMPLETED SUCCESSFULLY (NO ERROR FLAG) THEN HALT
                                ;
000060  032767  000040  177102  3$:     BIT #DONEBIT, RXCS      ; TEST FOR THE DONE FLAG
000066  001774                          BEQ 3$                  ; BEQ UNTIL THE DONE FLAG SETS
000070  005767  177074                  TST RXCS                ; TEST FOR THE ERROR FLAG
000074  001031                          BNE ERFLAG              ; BNE IF AN ERROR HAS OCCURED
000076  000000                          HALT                    ; OK = COMPLETED
                                ;THE ERROR FLAG IS SET
                                ;
                                ;THE CONTENTS OF THE RXES IS THE ERROR STATUS
                                ;
                                ;IF THE RXES BITS 1 AND 0 = 0 THEN SOME TYPE OF SEEK ERROR OCCURED
                                ;IF THE RXES BIT 0 = 1 THEN A CRC ERROR HAS OCCURED
                                ;IF THE RXES BIT 1 = 1 THEN A PARITY ERROR HAS OCCURED
                                ;
000100  032767  000003  177064  ERFLAG: BIT #3, RXES             ; TEST FOR CRC AND PARITY ERRORS
000106  001414                          BEQ SEEK                ; NOT A PARITY OR CRC (MUST) BE A SEEK
000110  032767  000002  177054          BIT #2, RXES            ; TEST FOR PARITY ERROR
000116  001404                          BEQ CRC                 ; NOT A PARITY ERROR (MUST) BE A CRC
                                ;
                                ;A PARITY ERROR HAS OCCURED
                                ;
                                ;INCREMENT AND TEST THE PARITY ERROR RETRY COUNTER PROGRAM LOCATION " PTRY "
                                ;
                                ;AND RETRY THE " COMMAND " UNTIL THE PARITY ERROR RECOVERS
                                ;
                                ;OR UNTIL THE PTRY COUNTER OVERFLOWS TO 0
                                ;
000120  005267  000202                  INC PTRY
000124  001336                          BNE RETRY               ; RETRY THE COMMAND
000126  000000                          HALT                    ; HARD PARITY ERROR
                                ;
                                ;A CRC ERROR HAS OCCURED
                                ;
                                ;INCREMENT AND TEST THE CRC ERROR RETRY COUNTER PROGRAM LOCATION " CTRY "
                                ;
                                ;AND RETRY THE COMMAND UNTIL THE CRC ERROR RECOVERS
                                ;
                                ;OR UNTIL THE CTRY COUNTER OVERFLOWS TO 0
                                ;
000130  005267  000174          CRC:    INC CTRY
000134  001332                          BNE RETRY               ; RETRY THE COMMAND
000136  000000                          HALT                    ; HARD CRC ERROR
                                ;THE ERROR FLAG IS SET
                                ;
                                ;THE ERROR IS (NOT) A PARITY ERROR AND IS (NOT) A CRC ERROR
                                ;
                                ;THEREFORE IT MUST BE A SEEK ERROR
                                ;
                                ; (STATE OF RXCS BITS 0 AND 1 ARE 0)
                                ;
000140  012767  040000  177022  SEEK:   MOV #INIT, RXCS          ; INITIALIZE
                                ;
                                ;INCREMENT AND TEST THE SEEK ERROR RETRY COUNTER PROGRAM LOCATION " STRY "
                                ;
                                ;AND RETRY THE COMMAND UNTIL THE SEEK ERROR RECOVERS
                                ;
                                ;OR UNTIL THE CTRY COUNTER OVERFLOWS TO 0
                                ;
000146  005267  000168                  INC STRY
000152  001323                          BNE RETRY               ; RETRY THE COMMAND
000154  000000                          HALT                    ; HARD SEEK ERROR
```

Figure 3-6 RX11 Write/Write Deleted Data/Read Example

```
160                                     ;THE FOLLOWING IS A PROGRAMMING EXAMPLE OF PROTOCOL REQUIRED TO
161                                     ;
162                                     ;EMPTY THE  SECTOR BUFFER OF 128 8-BIT BYTES
163                                     ;
164  000242  012767  177770  000056     ECNTRY: MOV #-10, PTRY          ; 8 TRYS TO EMPTY THE SECTOR BUFFER
165  000250  012700  000342             ESETUP: MOV #BUFFER, R0         ; PROGRAMS DATA BUFFER
166  000254  016767  000034  176706             MOV COMMAND, RXCS       ; ISSUE THE COMMAND
167                                     ;
168                                     ;WAIT FOR A TRANSFER REQUEST FLAG BEFORE TRANSFERRING DATA TO THE PROGRAMS
169                                     ;
170                                     ;DATA BUFFER FROM THE RX01 SECTOR BUFFER
171                                     ;
172                                     ;WAIT FOR A DONE FLAG TO INDICATE THE COMPLETION OF THE EMPTY BUFFER COMMAND
173                                     ;
174                                     ;PRIOR TO TESTIN G THE ERROR FLAG
175                                     ;
176  000262  105767  176702             ELOOP:  TSTB RXCS               ; TEST FOR TRANSFER REQUEST FLAG
177  000266  001014                              BMI EMPTY               ; BNE IF TRANSFER REQUEST FLAG IS SET
178  000270  032767  000040  176672              BIT #DONEBIT, RXCS      ; TEST FOR DONE FLAG
179  000276  001771                              BEQ ELOOP               ; BEQ UNTIL THE DONE FLAG SETS
180                                     ;
181                                     ;THE DONE FLAG IS SET
182                                     ;
183                                     ;TEST FOR ANY ERRORS (ONLY ERROR POSSIBLE IS A PARITY ERROR)
184                                     ;
185  000300  005767  176664                      TST RXCS
186  000304  001001                              BNE 1$
187  000306  000000                              HALT                    ; NO ERRORS = OK = COMPLETE
188                                     ;
189                                     ;INCFEMENT AND TEST THE PARITY ERROR RETRY PROGRAM LOCATION " PTRY "
190                                     ;
191                                     ;AND RETRY THE COMMAND UNTIL THE ERROR RECOVERS
192                                     ;
193                                     ;OR UNTIL THE PTRY CUNTER OVERFLOWS TO 0
194                                     ;
195  000310  005267  000012             1$:     INC PTRY
196  000314  001355                              BNE ESETUP              ; RETRY TO EMPTY THE SECTOR BUFFER
197  000316  000000                              HALT                    ; HARD PARITY ERROR
198                                     ;
199                                     ;THE TRANSFER REQUEST FLAG IS SET
200                                     ;
201                                     ;TRANSFER DATA TO THE PROGRAM  DATA BUFFER FROM THE RX01 SECTOR BUFFER
202                                     ;
203  000320  116738  176646             EMPTY:  MOVB RXDB, @(R0)+
204  000324  000756                              BR ELOOP
205
206                                     ;THE FOLLOWING 3 PROGRAM LOCATIONS ARE THE ERROR RETRY COUNTERS
207                                     ;
208  000326  000000                     PTRY:   0                       ; PARITY ERROR RETRY COUNTER
209  000330  000000                     CTRY:   0                       ; CRC ERROR RETRY COUNTER
210  000332  000000                     STRY:   0                       ; SEEK ERROR RETRY COUNTER
211                                     ;
212                                     ;PROGRAM LOCATION " COMMAND " CONTAINS THE COMMAND TO BE ISSUED VIA THE LCO IOT
213                                     ;
214                                     ;WRITE (4), WRITE DELETED DATA (14), OR READ (6), OR EMPTY BUFFER (2)
215                                     ;
216  000334  000000                     COMMAND:        0               ; 4, 14, 6, OR 2 + (GO BIT 1 = 1)
217                                     ;
218                                     ;PROGRAM LOCATION " SECTOR " CONTAINS THE SECTOR ADDRESS (1 TO 32 OCTAL)
219                                     ;
220  000336  000000                     SECTOR: 0                       ; 1 TO 32 OCTAL
221                                     ;
222                                     ;PROGRAM LOCATION " TRACK " CONTAINS THE TRACK ADDRESS (0 TO 114 OCTAL)
223                                     ;
224  000340  000000                     TRACK:  0                       ; 0 TO 114 OCTAL
225                                     ;
226                                     ;PROGRAM EQUIVALENTS
227                                     ;
228          000040                     DONEBIT=40
229          040000                     INIT=40000
230          000342                     BUFFER=.
231          000542                             .=BUFFER+200
232          000001                             .END
```

Figure 3-7 RX11 Empty Buffer Example

If no error is indicated, the program looks for the Transfer Request (TR) flag to set. The Error flag is retested if TR is not set. Once TR sets, a byte is moved from the RX11 sector buffer to the core locations of BUFFER. The process continues until the sector buffer is empty and the Done bit is set.

3.4.3 Fill Buffer Function

Figure 3-8 presents a program to implement a Fill Buffer function. It is very similar to the Empty Buffer example.

3.5 RESTRICTIONS AND PROGRAMMING PITFALLS

A set of restrictions and programming pitfalls for the RX11 is presented below.

1. Depending on how much data handling is done by the program between sectors, the minimum interleave of two sectors may be used, but to be safe a three-sector interleave is recommended.

2. If an error occurs and the program executes a Read Error Register function (111), a parity error may occur for that command. The error status would not be for the error in which the Read Error Register function was originally required.

3. The DRV SEL RDY bit is present only at the time of a Read Status function (101) for both drives, and after an Initialize, depending on the status of drive 0.

4. It is not required to load the Drive Select bit into the RXCS when the command is Fill Buffer (000) or Empty Buffer (010).

5. Sector Addressing: 1–26 (*No sector 0*)
   Track Addressing: 0–76

6. A power failure causing the recalibration of the drives will result in a Done condition, the same as finishing reading a sector. However, during a power failure, RXES bit 2 (Initialize Done) will set. Checking this bit will indicate a power fail condition.

7. Excessive usage of the Read Status function (101) will result in drastically decreased throughput, because a Read Status function requires between one and two diskette revolutions or about 250 ms to complete.

3.6 ERROR RECOVERY

There are two error indications given by the RX11 system. The Read Status function (Paragraph 3.3.5) will assemble the current contents of the RXES (Paragraph 3.2.2), which can be sampled to determine errors. The Read Error Register function (Paragraph 3.3.7) can also be used to retrieve explicit error information. The RX11 Interface register can be interrogated to determine the type of failure that occurred.

A list of error codes is presented on the following page.

NOTE

A Read Status function is not necessary if the DRV RDY bit is not going to be interrogated, because the RXES is in the Interface register at the completion of every function.

Octal Code	Error Code Meaning
0010	Drive 0 failed to see home on Initialize.
0020	Drive 1 failed to see home on Initialize.
0030	Found home when stepping out 10 tracks for INIT.
0040	Tried to access a track greater than 77.
0050	Home was found before desired track was reached.
0060	Self-diagnostic error.
0070	Desired sector could not be found after looking at 52 headers (2 revolutions).
0110	More than 40 μs and no SEP clock seen.
0120	A preamble could not be found.
0130	Preamble found but no I/O mark found within allowable time span.
0140	CRC error on what we thought was a header.
0150	The header track address of a good header does not compare with the desired track.
0160	Too many tries for an IDAM (identifies header).
0170	Data AM not found in allotted time.
0200	CRC error on reading the sector from the disk. No code appears in the ERREG.
0210	All parity errors.

```
                                        ;THE FOLLOWING IS A PROGRAMMING EXAMPLE OF THE PROTOCOL REQUIRED TO
                                        ;
                                        ;FILL THE SECTOR BUFFER WITH 128 8-BIT BYTES
                                        ;
                                        ; NOTE: THE DATA TO FILL THE SECTOR BUFFER CAN BE ASSEMBLED IN CORE IN THE
                                        ;       EVEN ADDRESSES BYTES OF 128 WORDS OR IN BOTH BYTES OF 64 WORDS
                                        ;
000156  012767  177770  000142  ENTRY: MOV #-10, PTRY          ; 8 TRYS TO FILL THE SECTOR BUFFER
000164  012700  000342          SETUP: MOV #BUFFER, R0          ; PROGRAMS DATA BUFFER
000170  016767  000140  176772         MOV COMMAND, RXCS        ; ISSUE THE COMMAND
                                        ;
                                        ;WAIT FOR A TRANSFER REQUEST FLAG BEFORE TRANSFERRING DATA FROM THE PROGRAMS
                                        ;
                                        ;DATA BUFFER TO THE RX01 SECTOR BUFFER
                                        ;
                                        ;WAIT FOR A DONE FLAT TO INDICATE THE COMPLETION OF THE FILL BUFFER COMMAND
                                        ;
                                        ;PRIOR TO TESTING THE ERROR FLAG
                                        ;
000176  105767  176766          LOOP:  TSTB RXCS               ; TEST FOR TRANSFER REQUEST FLAG
000202  001414                         BMI FILL                ; BEQ IF TRANSFER REQUEST FLAG SET
000204  032767  000040  176756         BIT #DONEBIT, RXCS      ; TEST FOR THE DONE FLAG
000212  001771                         BEQ LOOP                ; BEQ UNTIL THE DONE FLAG SETS
                                        ;
                                        ;THE DONE FLAG IS SET
                                        ;
                                        ;TEST FOR ANY ERRORS (ONLY ERROR POSSIBLE IS A PARITY ERROR)
                                        ;
000214  005767  176750                  TST RXCS
000220  001001                         BNE 1$
000222  000000                         HALT                    ; NO ERRORS = OK = COMPLETE
                                        ;
                                        ;INCREMENT AND TEST THE PARITY ERROR RETRY PROGRAM LOCATION " PTRY "
                                        ;
                                        ;AND RETRY THE COMMAND UNTIL THE ERROR RECOVERS
                                        ;
                                        ;OR UNTIL THE PTRY COUNTER OVERFLOWS TO 0
                                        ;
000224  005267  000076          1$:    INC PTRY
000230  001355                         BNE SETUP               ; RETRY TO FILL THE SECTOR BUFFER
000232  000000                         HALT                    ; HARD PARITY ERROR
                                        ;
                                        ;THE TRANSFER REQUEST FLAG IS SET
                                        ;
                                        ;TRANSFER DATA FROM THE PROGRAMS DATA BUFFER TO THE RX01 SECTOR BUFFER
                                        ;
000234  113867  176732          FILL:  MOVB @(R0)+, RXDB       ; PROGRAMS DATA BUFFER IS 64 WORDS IN LENGTH
000240  000756                         BR LOOP
```

Figure 3-8 RX11 Fill Buffer Example

CHAPTER 4
RX8E INTERFACE PROGRAMMING INFORMATION

The RX8E interface allows two modes of data transfer: 8-bit word length and 12-bit word length. In the 12-bit mode, 64 words are written in a diskette sector, thus requiring two sectors to store one page of information. The diskette capacity in this mode is 128,128 12-bit words (1,001 pages). In the 8-bit transfer mode, 128 8-bit words are written in each sector. Disk capacity is 256,256 8-bit words, which is a 33 percent increase in disk capacity over the 12-bit mode. Eight-bit mode must be used for generating IBM-compatible diskettes, since 12-bit mode does not fully pack the sectors with data. The hardware puts in the extra 0s. Data transfer requests occur 23 μs after the previous request was serviced for 12-bit mode (18 μs for 8-bit mode). There is no maximum time between the transfer request from the RX01 and servicing of that request by the host processor. This allows the data transfer to and from the RX01 to be interrupted without loss of data.

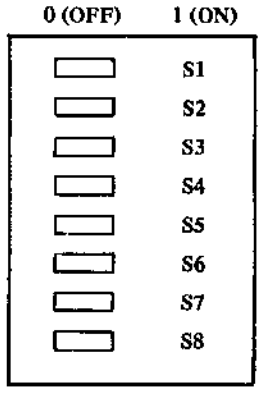
4.1 DEVICE CODES

The eight possible device codes that can be assigned to the interface are 70–77. These device codes define address locations of a specific device and allow up to eight RX8E interfaces to be used on a single PDP-8. These multiple device codes are also shared with other devices. Depending on what other devices are on the system, the RX8E device code can be selected to avoid conflicts. (Refer to the *PDP-8 Small Computer Handbook* for specific device codes.)

The device codes are selected by switches according to Table 4-1. These switches control AC bits 6–8, while AC bits 3–5 are fixed at 1s. The device code is initially selected to be 70. Switches 7 and 8 are not connected and will not affect the device selection code. The switches are all located on a single DIP switch package that is located on the M8357 RX8E interface board.

Table 4-1
Device Code Switch Selection

Device Code	S1	S2	S3	S4	S5	S6	S7	S8
77	0	0	0	1	1	1	X	X
76	0	0	1	1	1	0	X	X
75	0	1	0	1	0	1	X	X
74	0	1	1	1	0	0	X	X
73	1	0	0	0	1	1	X	X
72	1	0	1	0	1	0	X	X
71	1	1	0	0	0	1	X	X
70	1	1	1	0	0	0	X	X



4.2 INSTRUCTION SET

The RX8E instruction set is listed below and described in the following paragraphs.

IOT	Mnemonic	Description
67x0		No Operation
67x1	LCD	Load Command, Clear AC
67x2	XDR	Transfer Data Register
67x3	STR	Skip on Transfer Request Flag, 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

4.2.1 Load Command (LCD) – 67x1

This command transfers the contents of the AC to the Interface register and clears the AC. The RX01 begins to execute the function specified in AC 8, 9, and 10 on the drive specified by AC 7. A new function cannot be initiated unless the RX01 has completed the previous function. The command word is defined as shown in Figure 4-1.

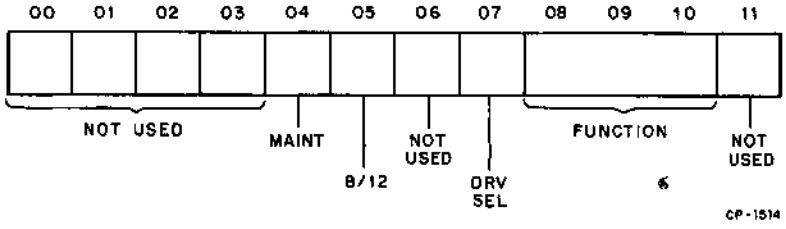


Figure 4-1 LCD Word Format (RX8E)

The command word is described in greater detail in Paragraph 4.3.1.

4.2.2 Transfer Data Register (XDR) – 67x2

With the Maintenance flip-flop cleared, this instruction operates as follows. A word is transferred between the AC and the Interface register. The direction of the transfer is governed by the RX01, and the length of the word transferred is governed by the mode selected (8-bit or 12-bit). When Done is negated, executing this instruction indicates to the RX01:

1. That the last data word supplied by the RX01 has been accepted by the PDP-8, and the RX01 can proceed, or

2. That the data or address word requested by the RX01 has been provided by the PDP-8, and the RX01 can proceed.

A data transfer (XDR) from the AC always leaves the AC unchanged. If operation is in 8-bit mode, AC 0–3 are transferred to the Interface register but are ignored by the RX01. Transfers into the AC are 12-bit jam transfers when in 12-bit mode. When in 8-bit mode, the 8-bit word is ORed into AC 4–11, and AC 0–3 remain unchanged. When the RX01 is done, this instruction can be used to transfer the RXES status word from the Interface register to the AC. The selected mode controls this transfer as indicated above.

4.2.3 STR – 67x3

This instruction causes the next instruction to be skipped if the Transfer Request (TR) flag has been set by RX01 and clears the flag. The TR flag should be tested prior to transferring data or address words with the XDR instruction to ensure the data or address has been received or transferred, or after an LCD instruction to ensure the command is in the Interface register. In cases where an XDR follows an LCD, the TR flag needs to be tested only once between the two instructions. (See programming example in Paragraph 4.5.1.)

4.2.4 SER – 67x4

This instruction causes the next instruction to be skipped if the Error flag has been set by an error condition in the RX01 and clears the flag. An error also causes the Done flag to be set (Paragraph 4.3.6).

4.2.5 SDN – 67x5

This instruction causes the next instruction to be skipped if the Done flag has been set by the RX01 indicating the completion of a function or detection of an error condition. If the Done flag is set, it is cleared by the SDN instruction. This flag will interrupt if interrupts are enabled.

4.2.6 INTR – 67x6

This instruction enables interrupts by the Done flag if AC 11 = 1. It disables interrupts if AC 11 = 0.

4.2.7 INIT – 67x7

The instruction initializes the RX01 by moving the head position mechanism of drive 1 (if drive 1 is available) to track 0. It reads track 1, sector 1 of drive 0. It zeros the Error and Status register and sets Done upon successful completion of Initialize. Up to 1.8 seconds may elapse before the RX01 returns to the Done state. Initialize can be generated programmably or by the Omnibus Initialize.

4.3 REGISTER DESCRIPTION

Only one physical register (the Interface register) exists in the RX8E, but it may represent one of the six RX01 registers described in the following paragraphs, according to the protocol of the function in progress.

4.3.1 Command Register (Figure 4-2)

The command is loaded into the Interface register by the LCD instruction (Paragraph 4.2.1).

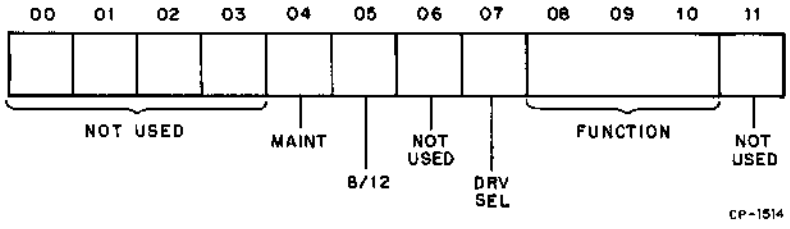


Figure 4-2 Command Register Format (RX8E)

The function codes (bits 8, 9, 10) are summarized below and described in Paragraph 4.4.

Code	Function
000	Fill Buffer
001	Empty Buffer
010	Write Sector
011	Read Sector
100	Not used
101	Read Status
110	Write Deleted Data Sector
111	Read Error Register

The DRV SEL bit (bit 7) selects one of the two drives upon which the function will be performed:

AC 7 = 0	Select drive 0
AC 7 = 1	Select drive 1

The 8/12 bit (bit 5) selects the length of the data word.

AC 5 = 0	Twelve-bit mode selected
AC 5 = 1	Eight-bit mode selected

The RX8E will initialize into 12-bit mode.

4.3.2 Error Code Register (Figure 4-3)

Specific error codes can be accessed by use of the Rear Error Register function (111) (Paragraph 4.4.7). The specific octal error codes are given in Paragraph 4.7.

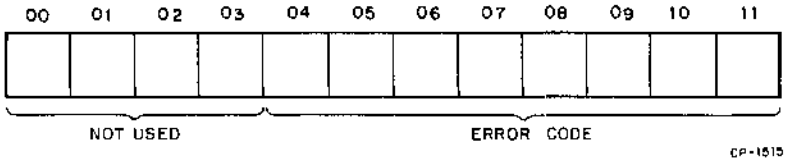


Figure 4-3 Error Code Register Format

The Maintenance bit (M bit) can be used to diagnose the RX8E interface under off-line and on-line conditions. The off-line condition exists when the BC05L-15 cable is disconnected from the RX01; the on-line condition exists when the cable is connected to the RX01.

If an LCD IOT (I/O Transfer) is issued with AC 4 = 1, the Maintenance flip-flop is set. When the Maintenance flip-flop is set, the assertion of RUN on *following* XDR instructions is inhibited, and all data register transfers (XDR) are forced into the AC. The Maintenance bit allows the Interface register to be written and read for maintenance checks. The Maintenance flip-flop is cleared by Initialize or by an LCD IOT with AC 4 = 0.

The following paragraphs describe more explicitly how to use the Maintenance bit in an off-line mode.

The contents of the interface buffer cannot be guaranteed immediately following the first LCD IOT, which sets the Maintenance flip-flop. However, successive LCD IOTs will guarantee the contents of the Interface register. The contents of the Interface register can then be verified by using the XDR IOT to transfer those contents into the AC.

In addition, the Maintenance flip-flop directly sets the Skip flags, which will remain set as long as the Maintenance flip-flop is set. Skipping in these flags as long as the Maintenance flip-flop is set will not clear the flags. Setting and then clearing the Maintenance flip-flop will leave the Skip flags in a set condition. The skip IOTs can then be issued to determine whether or not a large portion of the interface skip logic is working correctly.

The Maintenance flip-flop can also be used to determine if the interface is capable of generating an interrupt on the Omnibus. The Maintenance flip-flop is set, thus causing the Done flag to set. The Interrupt Enable flip-flop can be set by issuing an INTR IOT with AC bit 11 = 1. The combination of Done and Interrupt Enable should generate an interrupt.

The Maintenance flip-flop can also be used to test the INIT IOT. The Maintenance flip-flop is set and cleared to generate the flags, and INIT IOT is then executed. If execution of INIT IOT is internally successful, all of the flags and the Interrupt Enable flip-flop should be cleared if they were previously set.

In the on-line mode, use of the Maintenance bit should be restricted to writing and reading the Interface register. The same procedure described to write and read the Interface register in the off-line mode should be implemented in the on-line mode. Additional testing of the RX8E in the on-line mode should reference the appropriate circuit schematics. Exiting from the on-line Maintenance bit mode should be finalized by an initialize to the RX01.

4.3.3 RXTA – RX Track Address (Figure 4-4)

This register is loaded to indicate on which of the 77 tracks a given function is to operate. It can be addressed only under the protocol of the function in progress (Paragraph 4.4). Bits 0 through 3 are unused and are ignored by the control.

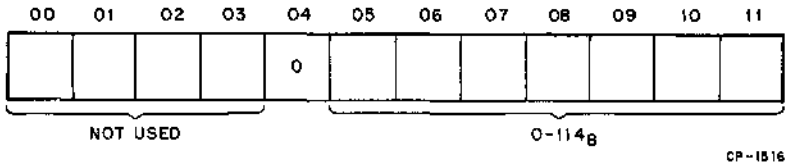


Figure 4-4 RXTA Format (RX8E)

4.3.4 RXSA – RX Sector Address (Figure 4-5)

This register is loaded to indicate on which of the 26 sectors a given function is to operate. It can be addressed only under the protocol of the function in progress (Paragraph 4.4). Bits 0 through 3 are unused and are ignored by the control.

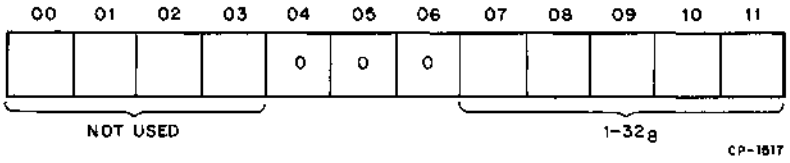


Figure 4-5 RXSA Format (RX8E)

4.3.5 RXDB – RX Data Buffer (Figure 4-6)

All information transferred to and from the floppy media passes through this register and is addressable only under the protocol of the function in progress. The length of data transfer is either 8 or 12 bits, depending on the state of bit 5 of the Command register when the LCD IOT is issued (Paragraph 4.3.1).

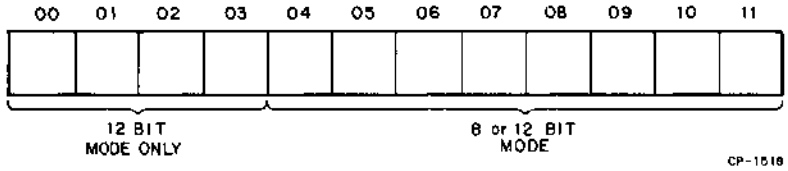


Figure 4-6 RXDB Format (RX8E)

4.3.6 RX Error and Status (Figure 4-7)

The RXES contains the current error and status conditions of the selected drive. This read-only register can be accessed by the Read Status function (101). The RXES is also available in the Interface register upon completion of any function. The RXES is accessed by the XDR instruction. The meaning of the error bits is given below.

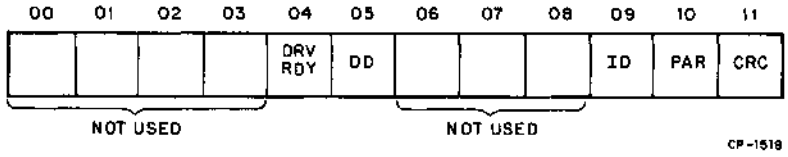


Figure 4-7 RXES Format (RX8E)

Bit No.	Description
11	CRC Error – The cyclic redundancy check at the end of the header or data field has indicated an error. The header or data must be considered invalid; it is suggested that the data transfer be retried up to ten times, as most data errors are recoverable (soft). (See Chapter 6).
10	Parity Error – When status bit 10 = 1, a parity error has been detected on command and address information being transferred to the RX01 μCPU controller from the RX8E interface. Upon detection of a parity error, the current function is terminated, the RXES status word is moved to the Interface register, and the Error and Done flags are set. The function can be retried to determine if the parity error is a soft or hard error. A parity error indication means that there is a problem in the interface cable between the RX01 and the interface.
9	Initialize Done – This bit indicates completion of the Initialize routine. It can be asserted due to RX01 power failure, system power failure, or programmable or bus Initialize. This bit is *not* available within the RXES from a Read Status function.

Bit No.	Description
5	Deleted Data (DD) – In the course of reading data, a deleted data mark was detected in the identification field. The data following will be collected and transferred normally, as the deleted data mark has no further significance within the RX01. Any alteration of files or actual deletion of data due to this mark must be accomplished by user software. This bit will be set if a successful or unsuccessful Write Deleted Data function is performed.
4	Drive Ready – This bit is asserted if the unit currently selected exists, is properly supplied with power, has a diskette installed properly, has its door closed, and has a diskette up to speed.

NOTE 1

This bit is only valid for either drive when retrieved via a Read Status function or for drive 0 upon completion of an Initialize.

NOTE 2

If the Error bit was set in the RXCS but Error bits are not set in the RXES, then specific error conditions can be accessed via a Read Error Register function.

4.4 FUNCTION CODE DESCRIPTION

RX8E functions are initiated by means of the Load Command IOT (LCD). The Done flag should be tested and cleared with the SDN instruction in order to verify that the RX8E is in the Done state prior to issuing the LCD instruction. Upon receiving an LCD instruction while in the Done state, the RX8E enters the Not Done state while the command is decoded. Each of the eight functions summarized below requires that a strict protocol be followed for the successful transfer of data, status, and address information. The protocol for each function is described in the following sections, and a summary table is presented below.

	AC			
Octal	8	9	10	Function
0	0	0	0	Fill Buffer
2	0	0	1	Empty Buffer
4	0	1	0	Write Sector
6	0	1	1	Read Sector
10	1	0	0	Not Used
12	1	0	1	Read Status
14	1	1	0	Write Deleted Data Sector
16	1	1	1	Read Error Register

NOTE

AC bit 11 is assumed to be 0 in the above octal codes, since AC bit 11 can be 0 or 1.

4.4.1 Fill Buffer (000)

This function is used to load the RX01 sector buffer from the host processor with 64 12-bit words if in 12-bit mode or 128 8-bit words if in 8-bit mode. This instruction only loads the sector buffer. In order to complete the transfer to the diskette, another function, Write Sector, must be performed. The buffer may also be read back by means of the Empty Buffer function in order to verify the data.

Upon decoding the Fill Buffer function, the RX01 will set the Transfer Request (TR) flag, signaling a request for the first data word. The TR flag must be tested and cleared by the host processor with the STR instructions prior to each successive XDR IOT (Paragraph 4.2.3). The data word can then be transferred to the Interface register by means of the XDR IOT. The RX01 next moves the data word from the Interface register to the sector buffer and sets the TR flag as a request for the next data word. The sequence above is repeated until the sector buffer has been loaded (64 data transfers for 12-bit mode or 128 data transfers for 8-bit mode). After the 64th (or 128th) word has been loaded into the sector buffer, the RXES is moved to the Interface register, and the RX01 sets the Done flag to indicate the completion of the function. It is, therefore, unnecessary for the host processor to keep a count of the data transfers. Any XDR commands after Done is set will result in the RXES status word being loaded in the AC. The sector buffer must be completely loaded before the RX8E will set Done and recognize a new command. An interrupt would now occur if Interrupt Enable were set.

4.4.2 Empty Buffer (001)

This function moves the contents of the sector buffer to the host processor. Upon decoding this function, RXES bits 10 and 11 (Parity Error and CRC Error) are cleared, and the TR flag is set with the first data word in the Interface register. This TR flag signifies the request for a data transfer from the RX8E to the host processor. The flag must be tested and cleared, then the word can be moved to the AC by an XDR command. The direction of the transfer for an XDR command is controlled by the RX01. The TR flag is set again with the next word in the Interface register. The above sequence is repeated until 64 words (128 bytes if 8-bit mode) have been transferred, thus emptying the sector buffer. The Done flag is then set after the RXES is moved in the Interface register to indicate the end of the function. An interrupt would now occur if Interrupt Enable were set.

NOTE

The Empty Buffer function does *not* destroy the contents of the sector buffer.

4.4.3 Write Sector (010)

This function transfers the contents of the sector buffer to a specific track and sector on the diskette. Upon decoding this function, the RX8E clears bits 10 and 11 (Parity Error and CRC Error) of the RXES and sets the TR flag, signifying a request for the sector address. The TR flag must be tested and cleared before the binary sector address can be loaded into the Interface register by means of the XDR command. The sector address must be within the limits $1-32_8$.

The TR flag is set, signifying a request for the track address. The TR flag must be tested and cleared, then the binary track address may be loaded into the Interface register by means of the XDR command. The track address must be within the limits $0-114_8$.

The RX01 tests the supplied track address to determine if it is within the allowable limits. If it is not, the RXES is moved to the Interface register, the Error and Done flags are set, and the function is terminated.

If the track address is legal, the RX01 moves the head of the selected drive to the selected track, locates the requested sector, transfers the contents of the sector buffer and a CRC character to that sector, and sets Done. Any errors encountered in the seek operation will cause the function to cease, the RXES to be loaded into the Interface register, and the Error and Done flags to be set. If no errors are encountered, the RXES is loaded into the Interface register and only the Done flag is set.

NOTE

The Write Sector function does *not* destroy the contents of the sector buffer.

4.4.4 Read Sector (011)

This function moves a sector of data from a specified track and sector to the sector buffer. Upon decoding this function, the RX8E clears RXES bits 5, 10, 11 (Deleted Data, Parity Error, CRC Error) and sets the TR flag, signifying the request for the sector address. The flag must be tested and cleared. The sector address is then loaded into the Interface register by means of the XDR command. The TR flag is set, signifying a request for the track address. The flag is tested and cleared by the host processor, and the track address is then loaded into the Interface register by an XDR command. The legality of the track address is checked by the RX01. If illegal, the Error and Done flags are set with the RXES moved to the Interface register, and the function is terminated. Otherwise, the RX01 moves the head to the specified track, locates the specified sector, transfers the data to the sector buffer, computes and checks CRC for the data. If no errors occur, the Done flag is set with the RXES in the Interface register. If an error occurs anytime during the execution of the function, the function is terminated by setting the Error and Done flags with RXES in the Interface register. A detection of CRC error results in RXES bit 11 being set. If a deleted data mark was encountered at the beginning of the desired data field, RXES bit 5 is set.

4.4.5 Read Status (101)

Upon decoding this function, the RX01 moves the RXES to the RX8E Interface register and sets the Done flag. The RXES can then be read by the Transfer Data Register command (XDR). The bits are defined in Paragraph 4.3.6.

NOTE

The average time for this function is 250 ms. Excessive use of this function will result in substantially reduced throughput.

4.4.6 Write Deleted Data Sector (110)

This function is identical to the Write Data function except that a deleted data mark is written prior to the data field rather than the normal data mark (Paragraph 1.3.3.2). RXES bit 5 (Deleted Data) will be set in the RX8E Interface register upon completion of the function.

4.4.7 Read Error Register Function (111)

The Read Error Register function can be used to retrieve explicit error information upon detection of the Error flag. Upon receiving this function, the RX01 moves an error code to the Interface register and sets Done. The Interface register can then be read via an XDR command and the code interrogated to determine which type of failure occurred (Paragraph 4.7).

NOTE

Care should be exercised in use of this function because, under certain conditions, erroneous error information may result (Paragraph 4.6).

4.4.8 Power Fail

There is no actual function code associated with Power Fail. When the RX01 senses a loss of power, it will unload the head and abort all controller action. All status signals are invalid while power is low.

When the RX01 senses the return of power, it will remove Done and begin a sequence to:

1. Move drive 1 head position mechanism to track 0.
2. Clear any active error bits.
3. Read sector 1 of track 1 of drive 0.
4. Set Initialize Done bit of the RXES, after which Done is again asserted.

There is no guarantee that information being written at the time of a power failure will be retrievable. However, all other information on the diskette will remain unaltered.

A method of aborting an incomplete function is with the INIT IOT (Paragraph 4.2.7).

4.5 PROGRAMMING EXAMPLES

4.5.1 Write/Write Deleted Data/Read Functions

Figure 4-8 presents a program for implementing a Write, Write Deleted Data, or a Read function with interrupts turned off (IOF). The first three steps preset the PTRY, CTRY, and STRY retry counters, which are set at ten retries but can be changed to any number. Starting at RETRY, the program tests for 8- or 12-bit mode, type of function, and drive. Once the command is loaded, the program waits in a loop for the controller to respond with Transfer Request (TR). When TR is set, the sector address is loaded and the AC is cleared. The program loops while waiting for the controller to respond with another TR. When TR is reset, the track address is loaded, and the AC is cleared again. The program loops to wait for the Done condition.

When the Done flag is set, the program checks for an error condition, indicated by the Error flag being set. If the AC = 0000, then the error is a seek error; if bit 10 of the AC is set, the error is a parity error; and if bit 11 of the AC is set, the error is a CRC error. Error status from the RXES is saved and tested to determine the error (Paragraph 4.3.6). The RXES will not include the Select Drive Ready bit. If a parity error is detected, the program increments and tests the PTRY retry counter. If a parity error persists after ten tries, it is considered a hard error. If ten retries have not occurred, a branch is made to RETRY and the sequence repeated.

If the Parity Error bit of the RXES is not set, then the program tests to see if the CRC Error bit is set. If a CRC error is detected, the program increments and tests the CTRY retry counter. If a CRC error persists after ten retries, it is considered a hard error. If ten retries have not occurred, a branch is made to RETRY and the sequence repeated.

A seek error is assumed if neither a CRC nor a parity error is detected. An Initialize (INIT) instruction is performed (Paragraph 4.2.7). During a Write or Write Deleted Data function, the sector buffer must be refilled, because INIT will cause sector 1 of track 1 of drive 0 to be read, which will destroy the previous contents of the sector buffer. The instruction sequence for a Fill Buffer function is not included in Figure 4-8, but is presented in Figure 4-10. After the system has been initialized, the program increments and tests the STRY retry counter. If a seek error persists after ten tries, it is considered a hard error. If ten retries have not occurred, a branch is made to RETRY and the sequence repeated.

4.5.2 Empty Buffer Function

Figure 4-9 shows a program for implementing an Empty Buffer function with interrupts turned off (IOF). The first instruction sets the number of retries at ten. A 2 is set in the AC to indicate an Empty Buffer command, and the command is loaded. When TR is set, the program jumps to EMPTY to transfer a word to the BUFFER location. A jump is made back to loop to wait for another TR. This process continues until either 64 words or 128 bytes have been emptied from the sector buffer. When Done is set, the program tests to see if the Error bit is set. If the Error bit is set, the program retries ten times. If the error persists, a hard parity error is assumed, indicating a problem in the interface cable.

4.5.3 Fill Buffer Function

Figure 4-10 presents a program to implement a Fill Buffer function. It is very similar to the Empty Buffer example.

```
                /PROGRAMMING EXAMPLES FOR THE RX8/RX01 FLEXIBLE DISKETTE
                /
                /THE FOLLOWING ARE RX01 IOT CODE DEFINITIONS
                /
                /THE STANDARD IOT DEVICE CODE IS 670-
                /
        6701    LCD=6701                /IOT TO LOAD THE COMMAND, (AC) IS THE COMMAND
        6702    XDR=6702                /IOT TO LOAD OR READ THE TRANSFER REGISTER
        6703    STR=6703                /IOT TO SKIP ON A TRANSFER REQUEST FLAG
        6704    SER=6704                /IOT TO SKIP ON AN ERROR FLAG
        6705    SDN=6705                /IOT TO SKIP ON THE DONE FLAG
        6706    INTR=6706               / (AC) = 0 INTERRUPT ENABLE OFF/ (AC) = 1 MEANS ON
        6707    INIT=6707               /IOT TO INITIALIAE THE RX8/RX01 SUBSYSTEM
                /
                /THE FOLLOWING IS A PROGRAMMING EXAMPLE OF THE PROTOCOL REQUIRED
                /
                /TO WRITE, WRITE DELETED DATA, OR READ AT SECTOR "S" (THE CONTENTS OF PROGRAM
                /
                /LOCATION SECTOR) OF TRACK "T" (THE CONTENTS OF PROGRAM LOCATION TRACK)
                /
                /IN 8 OR 12 BIT MODE
                /
0200  1254    START,  TAD KM10                                /-10
0201  3255            DCA PTRY                                /PARITY RETRY COUNTER
0202  1254            TAD KM10
0203  3256            DCA CTRY                                /CRC RETRY COUNTER
0204  1254            TAD KM10
0205  3257            DCA STRY                                /SEEK RETRY COUNTER
                /
                /WRITE, WRITE DELETED DATA, OR READ
                /
0206  1260    RETRY,  TAD MODE                                /0 IF 12-BIT, 100 IF 8-BIT
0207  1261            TAD COMMAND                             / 4 IF WRITE, 14 IF WRITE DELETED
                                                            /DATA, OR 6 IF READ
0210  1262            TAD UNIT                                / 0 IF UNIT 0, 20 IF UNIT 1
0211  6701            LCD                                     /IOT 67X1 TO LOAD THE COMMAND
                /
                /WAIT FOR THE TRANSFER REQUEST FLAG THEN TRANSFER THE SECTOR ADDRESS
                /
0212  6703            STR                                     /IOT 67X3 TO
0213  5212            JMP .-1                                 /WAIT FOR TRANSFER REQUEST FLAG
0214  1263            TAD SECTOR                              / 1 TO 32(OCTAL)
0215  6702            XDR                                     /IOT TO LOAD SECTOR
0216  7200            CLA                                     /CLA BECAUSE IOT XDR DOESN'T
                /
                /WAIT FOR THE TRANSFER REQUEST FLAG THEN TRANSFER THE TRACK ADDRESS
                /
0217  6703            STR                                     /IOT 67X3 TO
0220  5217            JMP .-1                                 /WAIT FOR TRANSFER REQUEST
0221  1264            TAD TRACK                               / 0 TO 114(OCTAL)
0222  6702            XDR                                     /IOT TO LOAD TRACK
0223  7200            CLA                                     /CLA BECAUSE IOT XDR DOESN'T

                /THE SECTOR AND TRACK ADDRESSES HAVE BEEN TRANSFERRED TO THE RX01 VIA THE XDR IOT
                /
                /WAIT FOR THE DONE FLAG AND CHECK FOR ANY ERRORS
                /
                /IF THE FUNCTION HAS COMPLETED SUCCESSFULLY (NO ERROR FLAG) THEN HALT
                /
0224  6705            SDN                                     /IOT 67X5 TO
0225  5224            JMP .-1                                 /WAIT FOR DONE FLAG
0226  6704            SER                                     /IOT 67X4 SAMPLES ERROR FLAG
0227  7402            HLT                                     / OK = COMPLETED
                /
                /THE ERROR FLAG IS SET
                /
                /THE CONTENTS OF THE TRANSFER REGISTER IS THE ERROR STATUS
                /
                /IF TRANSFER REGISTER BITS 10, AND 11 = 0 THEN SOME TYPE OF SEEK ERROR HAS OCCURED,
                /IF TRANSFER REGISTER BIT 11 = 1 THEN A CRC ERROR HAS OCCURED,
                /IF TRANSFER REGISTER BIT 10 = 1 THEN A PARITY ERROR HAS OCCURED
                /
0230  6702            XDR                                     /GET CONTENTS OF TR (ERROR STATUS)
0231  3265            DCA ASTATUS                             /AND SAVE
0232  7305            CLL CLA IAC RAL                         / 2
0233  0265            AND ASTATUS                             /TEST FOR PARITY ERROR
0234  7650            SNA CLA                                 /SKIP IF PARITY ERROR
0235  5241            JMP TCRC                                /NOT A PARITY ERROR - MAYBE CRC
                /
                /A PARITY ERROR HAS OCCURED
                /
                /INCREMENT AND TEST THE PARITY ERROR RETRY COUNTER PROGRAM LOCATION " PTRY "
                /
                /AND RETRY THE " COMMAND " UNTIL THE PARITY ERROR RECOVERS
                /
                /OR UNTIL THE PTRY COUNTER OVERFLOWS TO 0
                /
0236  2255            ISZ PTRY
0237  5206            JMP RETRY                               /RETRY THE COMMAND
0240  7402            HLT                                     /HARD PARITY ERROR
                /
                /THE ERROR FLAG IS SET BUT THE ERROR IS NOT A PARITY ERROR
                /
                /TEST FOR A CRC ERROR
                /
0241  7301    TCRC,   CLL CLA IAC                             / 1
0242  0265            AND ASTATUS                             /TEST FOR A CRC ERROR
0243  7650            SNA CLA                                 /SKIP IF A CRC ERROR
0244  5250            JMP SEEK                                /NOT A CRC - MUST BE A SEEK
```

Figure 4-8 RX8E Write/Write Deleted Data/Read Example (Sheet 1 of 2)

```
                        /A CRC ERROR HAS OCCURED
                        /
                        /INCREMENT AND TEST THE CRC ERROR RETRY COUNTER PROGRAM LOCATION " CTRY "
                        /
                        /AND RETRY THE COMMAND UNTIL THE CRC ERROR RECOVERS
                        /
                        /OR UNTIL THE CTRY COUNTER OVERFLOWS TO 0
                        /
0245  2256                      ISZ CTRY
0246  5206                      JMP RETRY                         /RETRY THE COMMAND
0247  7402                      HLT                               /HARD CRC ERROR
                        /
                        /THE ERROR FLAG IS SET
                        /
                        /THE ERROR IS [NOT] A PARITY ERROR AND IS [NOT] A CRC ERROR
                        /
                        /THEREFORE IS MUST BE A SEEK ERROR
                        /
                        / (CONTENTS OF THE TRANSFER REGISTER BITS 10, AND 11 = 0)
                        /
0250  6707    SEEK,     INIT                                      /IOT 67X7 TO INITIALIAE
                        /
                        /INCREMENT AND TEST THE SEEK ERROR RETRY COUNTER PROGRAM LOCATION " STRY "
                        /
                        /AND RETRY THE COMMAND UNTIL THE SEEK ERROR RECOVERS
                        /
                        /OR UNTIL THE CTRY COUNTER OVERFLOWS TO 0
                        /
0251  2257                      ISZ STRY
0252  5206                      JMP RETRY                         /RETRY THE COMMAND
0253  7402                      HLT                               /HARD SEEK ERROR

                        /THE FOLLOWING PROGRAM LOCATIONS ARE REFERENCED WITHIN THIS EXAMPLE
                        /
0254  7770    KM10,     -10
                        /
                        /THE FOLLOWING 3 PROGRAM LOCATIONS ARE THE ERROR RETRY COUNTERS
                        /
0255  0000    PTRY,     0                                         /PARITY ERROR RETRY COUNTER
0256  0000    CTRY,     0                                         /CRC ERROR RETRY COUNTER
0257  0000    STRY,     0                                         /SEEK ERROR RETRY COUNTER
                        /
                        /PROGRAM LOCATION " MODE " CONTAINS A 0 IF 12-BIT MODE, OR
                        /CONTAINS A 100 IF 8-BIT MODE
                        /
0260  0000    MODE,     0                                         / 0 OR 100
                        /
                        /PROGRAM LOCATION " COMMAND " CONTAINS THE COMMAND TO BE ISSUED VIA THE LCD IOT
                        /
                        /WRITE (4), WRITE DELETED DATA (14), OR READ (6), OR EMPTY BUFFER (2)
                        /
0261  0000    COMMAND,  0                                         / 4, 14, OR 6, OR 2
                        /
                        /PROGRAM LOCATION " UNIT " CONTAINS THE UNIT DESIGNATION
                        /
                        /UNIT 0 (0), OR UNIT 1 (20)
                        /
0262  0000    UNIT,     0                                         / 0, OR 20
                        /
                        /PROGRAM LOCATION " SECTOR " CONTAINS THE SECTOR ADDRESS (1 TO 32 OCTAL)
                        /
0263  0000    SECTOR,   0                                         / 1 TO 32 OCTAL
                        /
                        /PROGRAM LOCATION " TRACK " CONTAINS THE TRACK ADDRESS (0 TO 114 OCTAL)
                        /
0264  0000    TRACK,    0                                         / 0 TO 114 OCTAL
                        /
                        /PROGRAM LOCATION " ASTATUS " CONTAINS THE CONTENTS OF THE TRANSFER REGISTER
                        /
                        /AT THE DETECTION OF AN ERROR (ERROR FLAG = 1) WHICH CORRESPONDS TO THE
                        /
                        /ERROR STATUS
                        /
                        / = 0 IF SEEK ERROR, 1 IF CRC ERROR, 2 IF PARITY ERROR
                        /
0265  0000    ASTATUS,  0                                         /STATUS AT ERROR
```

Figure 4-8 RX8E Write/Write Deleted Data/Read Example (Sheet 2 of 2)

```
                        /THE FOLLOWING IS A PROGRAMMING EXAMPLE OF PROTOCOL REQUIRED TO
                        /
                        /EMPTY THE SECTOR BUFFER OF 64 12-BIT WORDS (12 BIT MODE), OR
                        /
                        /EMPTY THE SECTOR BUFFER OF 128 8-BIT BYTES (8 BIT MODE)
                        /
0312  1254      EENTRY, TAD KM10                        / 8 TRYS TO EMPTY THE SECTOR BUFFER
0313  3255            DCA PTRY                          /PARITY ERROR RETRY COUNTER
0314  1377      ESETUP, TAD (BUFFER-1)                  /PROGRAMS DATA BUFFER
0315  3010            DCA A10                           /AUTO INDEX REGISTER 10
0316  1260            TAD MODE                          / 0 IF 12-BIT, 100 IF 8 BIT
0317  1261            TAD COMMAND                       / 2 MEANS EMPTY BUFFER
0320  6701            LCD                               /IOT TO ISSUE THE COMMAND
                        /
                        /WAIT FOR A TRANSFER REQUEST FLAG BEFORE TRANSFERRING DATA TO THE PROGRAMS
                        /
                        /DATA BUFFER FROM THE RX01 SECTOR BUFFER
                        /
                        /WAIT FOR A DONE FLAG TO INDICATE THE COMPLETION OF THE EMPTY BUFFER COMMAND PRIOR TO
                        /
                        /TESTING THE ERROR FLAG
                        /
0321  6703      ELOOP,  STR                             /TEST FOR TR FLAG
0322  7410            SKP                               /TR NOT SET, TEST FOR DONE FLAG
0323  5333            JMP EMPTY                         /TR FLAG SET
0324  6705            SDN                               /TEST FOR DONE FLAG
0325  5274            JMP ELOOP                         /NOT TR, OR DONE YET
                        /
                        /THE DONE FLAG IS SET
                        /
                        /TEST FOR ANY ERRORS (ONLY ERROR POSSIBLE IS A PARITY ERROR)
                        /
0326  6704            SER                               /TEST FOR THE ERROR FLAG
0327  7402            HLT                               /NO ERRORS - OK
                        /
                        /INCREMENT AND TEST THE PARITY ERROR RETRY PROGRAM LOCATION " PTRY "
                        /
                        /AND RETRY THE COMMAND UNTIL THE ERROR RECOVERS
                        /
                        /OR UNTIL THE PTRY COUNTER OVERFLOWS TO 0
                        /
0330  2255            ISZ PTRY
0331  5314            JMP ESETUP                        /RETRY TO EMPTY THE SECTOR BUFFER
0332  7402            HLT                               /HARD PARITY ERROR
                        /
                        /THE TRANSFER REQUEST FLAG IS SET
                        /
                        /TRANSFER DATA TO THE PROGRAMS DATA BUFFER FROM THE RX01 SECTOR BUFFER
                        /
0333  6702      EMPTY,  XDR                             /FROM THE RX01 SECTOR BUFFER
0334  3410            DCA I A10                         /TO THE PROGRAMS DATA BUFFER
0335  5321            JMP ELOOP                         /LOOP UNTIL THE DONE FLAG SETS
0377  0377
      0400            PAGE
                        /THE FOLLOWING PROGRAM LOCATIONS ARE RESERVED FOR THE PROGRAMS DATA BUFFER
                        /
0400  0000      BUFFER, 0
      0600            *BUFFER+200
                        $
```

Figure 4-9 RX8E Empty Buffer Example

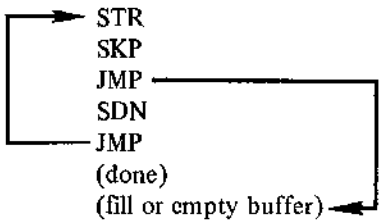
```
                        /THE FOLLOWING IS A PROGRAMMING EXAMPLE OF PROTOCOL REQUIRED TO
                        /
                        /FILL THE SECTOR BUFFER WITH 64 12-BIT WORDS (12 BIT MODE), OR
                        /
                        /FILL THE SECTOR BUFFER WITH 128 8-BIT BYTES (8 BIT MODE)
                        /
        0010    A10=10
                        /
0266    1254    FENTRY, TAD XM10                        / 0 TRYS TO FILL THE SECTOR BUFFER
0267    3255            DCA PTRY                        /PARITY ERROR RETRY COUNTER
0270    1377    SETUP,  TAD (BUFFER-1)                  /PROGRAMS DATA BUFFER
0271    3010            DCA A10                         /AUTO INDEX REGISTER 10
0272    1260            TAD MODE                        / 0 IF 12-BIT, 100 IF 8 BIT
0273    6701            LCD                             /IOT TO ISSUE THE COMMAND
                        /
                        /WAIT FOR A TRANSFER REQUEST FLAG BEFORE TRANSFERRING DATA FROM THE PROGRAMS
                        /
                        /DATA BUFFER TO THE RX01 SECTOR BUFFER
                        /
                        /WAIT FOR A DONE FLAG TO INDICATE THE COMPLETION OF THE FILL BUFFER COMMAND PRIOR TO
                        /
                        /TESTING THE ERROR FLAG
                        /
0274    6703    LOOP,   STR                             /TEST FOR TR FLAG
0275    7410            SKP                             /TR NOT SET, TEST FOR DONE FLAG
0276    5306            JMP FILL                        /TR FLAG SET
0277    6705            SDN                             /TEST FOR DONE FLAG
0300    5274            JMP LOOP                        /NOT TR, OR DONE YET
                        /
                        /THE DONE FLAG IS SET
                        /
                        /TEST FOR ANY ERRORS (ONLY ERROR POSSIBLE IS A PARITY ERROR)
                        /
0301    6704            SER                             /TEST FOR THE ERROR FLAG
0302    7402            HLT                             /NO ERRORS - OK
                        /
                        /INCREMENT AND TEST THE PARITY ERROR RETRY PROGRAM LOCATION " PTRY "
                        /
                        /AND RETRY THE COMMAND UNTIL THE ERROR RECOVERS
                        /
                        /OR UNTIL THE PTRY COUNTER OVERFLOWS TO 0
                        /
0303    2255            ISZ PTRY
0304    5270            JMP SETUP                       /RETRY TO FILL THE SECTOR BUFFER
0305    7402            HLT                             /HARD PARITY ERROR
                        /
                        /THE TRANSFER REQUEST FLAG IS SET
                        /
                        /TRANSFER DATA FROM THE PROGRAMS DATA BUFFER TO THE RX01 SECTOR BUFFER
                        /
0306    1410    FILL,   TAD I A10                       /VIA AUTO INDEX REGISTER 10
0307    6702            XDR                             /TO THE RX01 SECTOR BUFFER
0310    7200            CLA                             /CLA BECAUSE IOT XDR DOESN'T
0311    5274            JMP LOOP                        /LOOP UNTIL THE DONE FLAG SETS
```

Figure 4-10 Fill Buffer Example

4.6 RESTRICTIONS AND PROGRAMMING PITFALLS

A set of 11 restrictions and programming pitfalls for the RX8E is presented below.

1. When performing the following sequence of instructions, interrupts *must* be off.



   If interrupts are not off, the following sequence of events will occur. Assume interrupts are enabled and the RX8E issues an interrupt request just before the SDN instruction. The SDN instruction will be executed as the last legal instruction before the processor takes over. However, since the Done flag is cleared by the SDN instruction, the processor will not find the device that issued the interrupt.

2. The program must issue an SER instruction to test for errors following an SDN instruction.

3. For maximum data throughput for consecutive writes or reads in 8-bit mode, interleave every three sectors; in 12-bit mode, interleave every two sectors. (This of course depends on program overhead.)

4. When issuing the IOT XDR at the end of a function to test the status, the instruction AND 377 must be given, because the most significant bits (0–3) contain part of the previous command word.

5. If an error occurs and the program executes a Read Error Register function (111) (Paragraph 4.4.7), a parity error may occur for that command. The error code coming back would not be for the original error in which the Read Error Register function was issued, but for the parity error resulting from the Read Error Register function. Therefore, check for parity error with the Read Status function (101) before checking for errors with the Read Error Register function (111).

6. The SEL DRV RDY bit is present only at the time of the Read Status function (101) for either drive, or at completion of an Initialize for drive 0.

7. It is not necessary to load the Drive Select bit into the command word when the command is Fill Buffer (000) or Empty Buffer (001).

8. Sector Addressing: 1–26 or $1–32_8$ (*No sector 0*)
   Track Addressing: 0–76 or $1–114_8$

9. If a Read Error Register function (111) is desired, the program must perform this function before a Read Status function (101), because the content of the Error register is always modified by a Read Status function.

10. The instructions STR, SDN, SER also clear the respective flags after testing, so that the software must store these flags if future reference to them is needed after performing one of these instructions.

11. Excessive use of the Read Status function (101) will result in drastically decreased throughput, because a Read Status function requires between one and two diskette revolutions or about 250 ms to complete.

4.7 ERROR RECOVERY

There are two error indications given by the RX8E system. The Read Status function (Paragraph 4.4.5) will assemble the current contents of the RXES (Paragraph 4.3.6), which can be sampled to determine errors. The Read Error Register function (Paragraph 4.4.7) can also be used to retrieve explicit error information.

The results of the Read Status function or the Read Error Register function are in the Interface register when Done sets, indicating the completion of the function. The XDR IOT must be issued to transfer the contents of the Interface register to the PDP-8's AC.

NOTE

A Read Status function is not necessary if the DRV RDY bit is not going to be interrogated, because the RXES is in the Interface register at the completion of every function.

The error codes for the Read Error Register function are presented below.

Octal Code	Error Code Meaning
0010	Drive 0 failed to see home on Initialize.
0020	Drive 1 failed to see home on Initialize.
0030	Found home when stepping out 10 tracks for INIT.
0040	Tried to access a track greater than 77.
0050	Home was found before desired track was reached.
0060	Self-diagnostic error.
0070	Desired sector could not be found after looking at 52 headers (2 revolutions).
0110	More than 40 μs and no SEP clock seen.
0120	A preamble could not be found.
0130	Preamble found but no I/O mark found within allowable time span.
0140	CRC error on what we thought was a header.
0150	The header track address of a good header does not compare with the desired track.
0160	Too many tries for an IDAM (identifies header).
0170	Data AM not found in allotted time.
0200	CRC error on reading the sector from the disk. No code appears in the ERREG.
0210	All parity errors.

CHAPTER 5
THEORY OF OPERATION

This chapter presents a discussion of the hardware and μCPU firmware of the RX11 and RX8 Floppy Disk Systems. This information, combined with the programming information and functional descriptions contained in Chapters 3 and 4, should give the reader a complete knowledge of the theory of operation of the RX11 and RX8 Floppy Disk Systems.

The first section of this chapter describes the overall system block diagram and the signals that interconnect each of the blocks. The second section presents a detailed block diagram and logic discussion of each of the system blocks. The μCPU microprogram is discussed in Paragraphs 5.2.4 and 5.2.5.

5.1 OVERALL SYSTEM BLOCK DIAGRAM

The floppy disk system consists of four elements (Figure 5-1):

1. Drive mechanics, which includes actuators and transducers (up to two per controller)
2. Read/write electronics, which translates power levels between drive mechanics and control logic
3. Microprogrammed controller, which includes all control logic.
4. Bus interface, which translates between host processor bus protocol (Unibus or Omnibus) and RX01 data bus

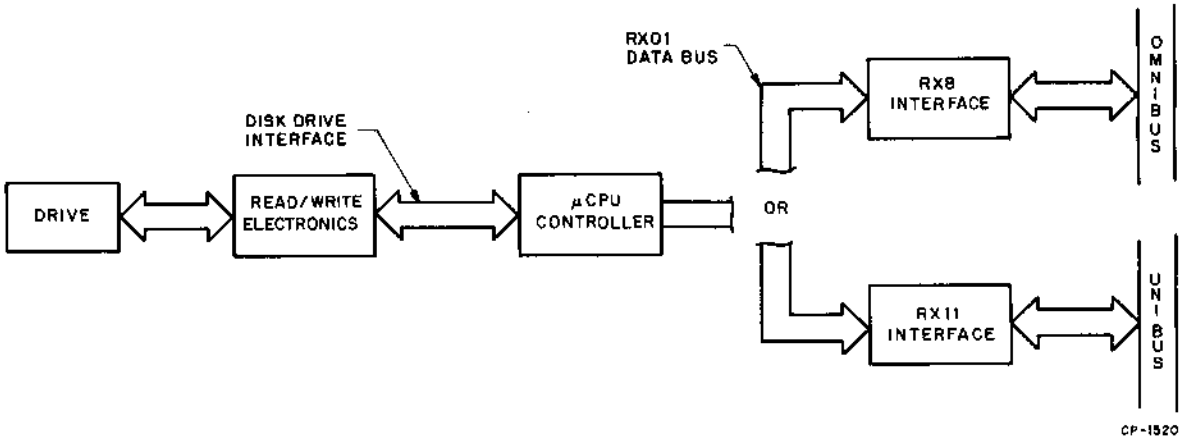


Figure 5-1 Bus Structure

There are three levels of data transmission in the floppy disk system (Figure 5-1):

1. Unibus for PDP-11 or Omnibus for PDP-8 for data transmission between bus interface and host processor

2. RX01 data bus for data transmission between the RX01 μCPU controller and the bus interface

3. The disk drive interface for data and control information transmission between the read/write electronics and the RX01 μCPU controller.

Signals between the read/write electronics and mechanical drive are analog signals used to control head motion and sense diskette motion and position. The sections which follow describe the signals used in the three levels of data transmission and the analog signals between the read/write electronics and mechanical drive.

5.1.1 Omnibus to RX8E Interface Signals

The RX8E interface communicates with the PDP-8 Omnibus via the signals shown in Figure 5-2 and described below. These signals are further explained in the *PDP-8 Small Computer Handbook.*

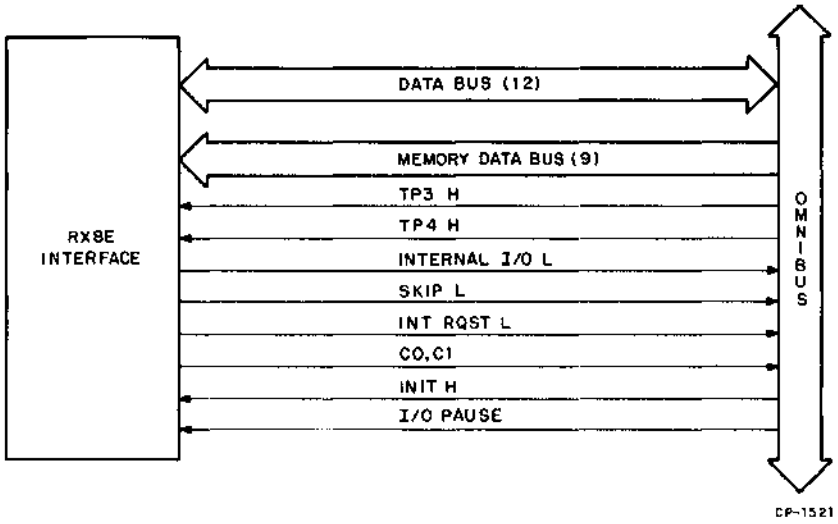


Figure 5-2 Omnibus to RX8E Interface Signals

DATA BUS – Twelve parallel bits of data are transferred along a bidirectional bus for both input and output data between the AC register in the processor and the Interface register in the RX8E interface.

MEMORY DATA BUS – This signal provides I/O transfer (IOT) instructions from memory to the RX8E interface.

TP3 H, TP4 H – These signals are used to clear the flag and clock the Interface register of the RX8E interface in transferring data along the data bus.

INTERNAL I/O L – This signal is grounded by the RX8E interface selector decoder to inhibit decoding any internal Omnibus I/O transfer (IOT) instructions. Failure to ground this line will result in long IOT timing.

SKIP L – An IOT checks the flag for a ONE state. If the flag is set, SKIP L is asserted and the address of the program counter (PC) plus one is loaded into the Central Processor Memory Address (CPMA) register to implement a skip.

INT RQST L – This signal is part of the interrupt structure of the Omnibus. It is the method by which the RX8E interface signals the processor that it has data to be serviced.

C0, C1 – Signals C0 and C1 determine the type of transfer between the RX8E interface and the processor. These signals control the data path within the processor and determine if data is to be placed on the data bus or received from the data bus. They are also used to develop the necessary load control signals required to load either the accumulator (AC) or the program counter (PC) in the processor.

INIT H – INIT H is a signal used to clear all flags in the RX8E interface and initialize the RX01.

I/O PAUSE L – This signal is used to gate the RX8E select and operation codes into the programmed I/O interface of the PDP-8 decoders.

5.1.2 Unibus to RX11 Interface Signals

The RX11 interface communicates with the PDP-11 Unibus via the signals shown in Figure 5-3 and described below. These signals are further explained in the *PDP-11 Peripherals Handbook.*

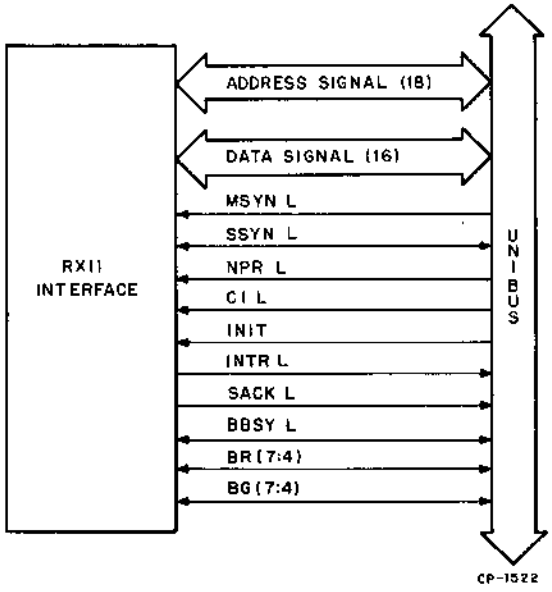


Figure 5-3 Unibus to RX11 Interface Signals

ADDRESS (A Lines) – The 18 address lines are used by the CPU to select the device register addresses of the RX11, which are 177170 (RXCS) and 177172 (RXDB).

DATA (D Lines) – The 16 parallel data lines are used to transfer information in and out of the RX11 interface.

MSYN L – This signal is the master synchronization control signal that is initiated by the RX11 when it has control of the Unibus for data transmission.

SSYN L – This signal is the slave synchronization control signal that is initiated by the RX11 in response to an MSYN L signal from the processor or another device that has control of the Unibus and is about to send data to the RX11.

NPR L – This signal from the processor will inhibit the RX11 interface from issuing a bus grant.

NOTE
The RX11 is *not* an NPR device.

C1 L – This bus signal is coded by the master device to control the slave in Data In mode (passing data to the Unibus) if it is negated and Data Out mode (passing data from the Unibus) if it is asserted.

INIT L – This is the signal asserted by the processor when the START key on the console is depressed, when a RESET instruction is executed, or when the power fail sequence occurs. This signal will initialize the RX11 system.

INTR L – This signal is asserted by the RX11 when it has bus control during an interrupt sequence. It directs the processor to go to interrupt service routine.

SACK L – This signal is sent by the RX11 to the processor in acknowledgment of Unibus control being transferred to it. This signal inhibits further bus grants by the processor.

BBSY L – This is the signal sent by the RX11 when asserting master control of the Unibus. This signal follows the SACK L signal.

BR (7:4) – These four priority bus request lines are used by the RX11 to request bus mastership. Each device of the same priority level passes a grant signal to the next device on the line, unless it has requested bus control; in this case, the requesting device blocks the signal from the following devices and assumes bus control.

BG (7:4) – These are four priority bus grant lines corresponding to the four request lines. The processor uses them to respond to a specific bus request.

5.1.3 Interface to μCPU Controller Signals

The μCPU controller and RX8E or RX11 interface communicate via the signals shown in Figure 5-4 and described below.

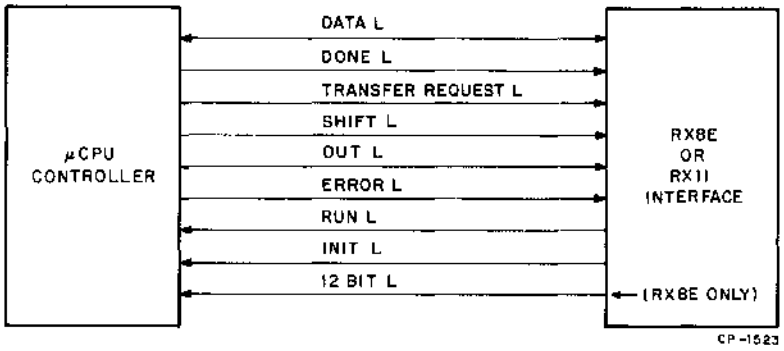


Figure 5-4 Interface to μCPU Controller Signals

INIT L – The RX01 will negate DONE L and move the head position mechanism of drive 1 (if it exists) to track 0. The RX01 will also read sector 1 of track 1 of drive 0 and then assert DONE L without error upon successful completion of the function.

DONE L – Asserted low, DONE L indicates that there is no RX01 function in progress. Initiating any function will cause DONE L to go false for the duration of that function. Attempting to initiate any function other than Initialize while DONE L is false is illegal and may result in an error.

RUN L – RUN L initiates communication between interface and controller. RUN L, asserted while DONE L is true, passes a command from interface to controller serially. DONE L will go false until the command has been executed (or until Initialize is asserted). RUN L, asserted while DONE L is false, signals transfer of data to or from the controller. All control lines to the controller must be stable 75 ns before RUN L is asserted.

OUT L – OUT L signals the direction in which the RX01 is prepared to transfer data. When OUT L is asserted, the direction of transfer is from controller to interface. When OUT L is negated, the direction is from interface to controller. OUT L is never asserted while DONE L is true, and OUT L is negated by Initialize.

TRANSFER REQUEST L (TR L) – The TR L line, with RUN L and OUT L, forms a bidirectional handshake set. On transfers from controller to interface (OUT L asserted), TR L going true indicates that the next data element has been transferred to the Interface register. The transfer of the following data element will be initiated by asserting RUN L. This will negate TR L until the new data element has been assembled in the interface.

On transfers from interface to controller (OUT L negated), TR asserted indicates that the controller is prepared to accept the next element of data. The arrival of the new data element will be signaled by assertion of RUN L. Assertion of RUN L while TR L is negated is an error.

DATA L – DATA L is a bidirectional line for transfer of data to and from the controller. A parity bit is appended to the serial data stream by the interface when the direction of the data transfer is into the controller. The controller will interrogate the parity bit for validity.

SHIFT L – The SHIFT L pulse strobes information to or from the controller bit-by-bit.

1. Interface to Controller Transfer – The transfer of either commands or data words begins with assertion of RUN L. Following the assertion of RUN L, DONE L or TR L will be negated and a number of SHIFT L pulses will occur. The number depends on the length of the data element to be passed. The first bit of data (or command) must be stable when RUN L is asserted. The SHIFT L pulse width is 200 ns nominal. SHIFT L pulses will not occur more often than every 400 ns. Subsequent bits of data may be brought up on the trailing edge of SHIFT L. DONE L or TR L will be reasserted following the last SHIFT L pulse.

2. Controller to Interface Transfer – The assertion of TR L indicates the controllers readiness to transfer data. Assertion of RUN L will negate TR L and initiate a train of SHIFT L pulses. The data is to be sampled on the leading edge of SHIFT L and is valid only while SHIFT L is asserted. TR L will be reasserted at the end of each element of data. DONE L will be asserted following transfer of the last element of data in a block.

12 BIT L – This signal is asserted by the interface to controller and determines the number of shift pulses generated by the controller for each byte transferred.

Signal 12 BIT L asserted will produce 12 SHIFT L pulses for data transfer between the interface and controller upon the assertion of RUN L. Signal 12 BIT L negated will produce eight SHIFT L pulses for data transfer between the interface and controller upon the assertion of RUN L. This line must remain asserted throughout the entire data transfer. When data is transferred, the most significant bit is transferred first.

NOTE

Signal 12 BIT L is *only* asserted by the RX8E interface for PDP-8 12-bit words. It is *never* asserted by the RX11 interface.

ERROR L – This is an error summary bit generated by the controller that sets when any error is detected (Paragraphs 3.2.1 and 4.3.6). The detection of ERROR L stops all controller action and asserts DONE L and the Error flag. This line is cleared by INIT L or the initiation of a new function.

5.1.4 μCPU Controller to Read/Write Electronics Signals

The μCPU controller and read/write electronics communicate via the signals shown in Figure 5-5 and described below.

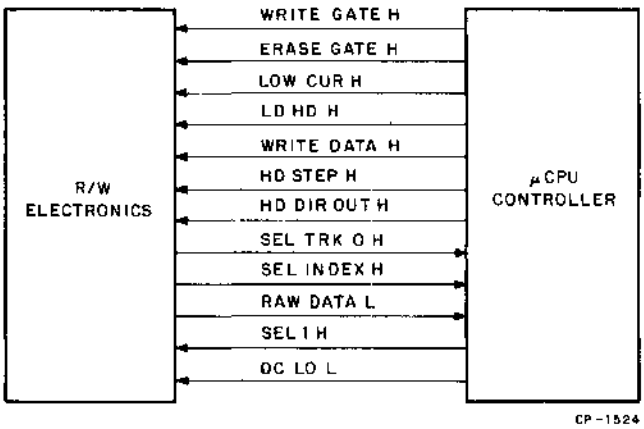


Figure 5-5 μCPU Controller to Read/Write Electronics Signals

LOW CUR H – This signal is asserted by the controller to select the lower of two write current levels when operating on a track above 43. As the head moves closer to the center of the disk, the bit density increases as linear velocity decreases, necessitating a reduction in write current.

WRITE DATA H – This signal conveys the complete data stream to the read/write electronics at TTL logic levels. Each transition on this line results in a flux reversal on the disk.

In general, the pattern will be one clock transition every 4 μs with an intervening transition between two successive clocks to represent a data one and no intervening transition to represent a data zero. It should be noted that the data content of this stream cannot be inferred from its instantaneous logic level, but only from the pattern of its transitions (Paragraph 1.3.2).

RAW DATA L – This signal conveys the complete data stream recovered from the diskette at TTL logic levels. It includes a regular pattern of clock transitions which the controller will separate from the data transitions. As above, the data content is in the pattern of transitions rather than the absolute level at any instant of time (Paragraph 1.3.2).

SEL 1 H – This signal uniquely selects one of the two possible diskette drives. The assertion of this line will select logical drive 1 for use. Unit 0 is physically the left-hand unit in the rack.

WRITE GATE H – This signal is asserted by the controller to enable the selected write drivers. This level must be asserted prior to the beginning of the data field to be written and is negated after the last bit of the data field. This timing is completely under microprogram control.

ERASE GATE H – This signal is used in conjunction with WRITE GATE H to enable the tunnel erase drivers. It is asserted and negated after the assertion of WRITE GATE H, with timing determined by the microprogrammed controller.

LD HD H – This signal is asserted by the controller to hold the media against the selected head.

HD STEP L – This signal is asserted twice by the controller to change head position by one track in a direction determined by signal HD DIR OUT H. The maximum step rate is 10 ms per step. Minimum pulse width is 200 ns.

HD DIR OUT H – This signal determines the direction in which the head will move in response to an HD STEP L signal. If HD DIR OUT H is unasserted, the heads will travel toward the center of the disk (IN), increasing the track address. If HD STEP L is asserted when HD DIR OUT H is asserted, the heads will travel toward the outside edge of the disk (OUT), decreasing the track address.

SEL TRK 0 H – This signal is asserted by the selected drive to indicate that its head is positioned over track 0.

SEL INDEX H – This signal is asserted by the selected drive to indicate that the hard index hole has been detected. This occurs once per revolution and is used by the control to time operations and detect "up to speed." This pulse is 400 μs minimum width.

DC LO L – This signal is asserted by an Initialize signal from the controller to the drives.

5.1.5 Read/Write Electronics to Drive Signals

The read/write electronics and drive(s) communicate through five sets of signals per drive as shown in Figure 5-6 and described below. The plug designations for the cabling are also shown in Figure 5-6.

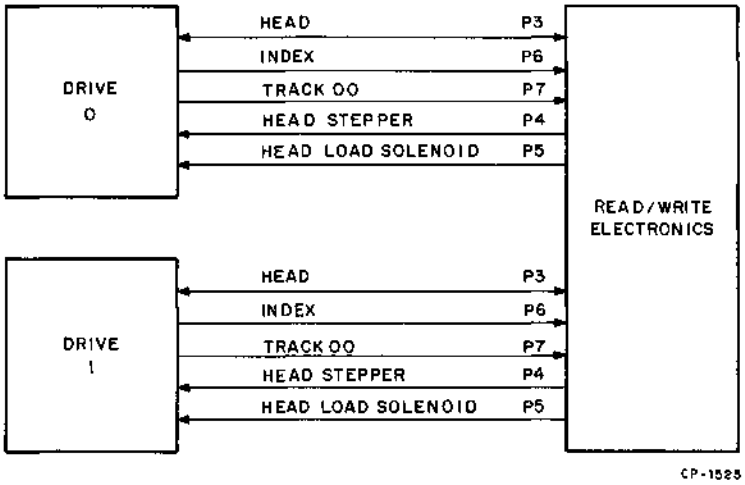


Figure 5-6 Read/Write Electronics to Drive Signals

HEAD – This is an analog signal to and from the drive head.

INDEX – This is a set of signals connected to a LED-phototransistor pair which locates the index hole for determination of diskette rotational position and speed.

TRACK 00 – This is a set of signals connected to a LED-phototransistor pair, which indicates positioning at track 0.

HEAD STEPPER – This signal is output from the read/write electronics, which moves the head from track to track.

HEAD LOAD SOLENOID – These signals activate a solenoid to load the head onto the diskette during a read/write operation. The head is unloaded from the diskette to reduce diskette wear when not performing a read/write operation.

5.2 DETAILED BLOCK DIAGRAM AND LOGIC DISCUSSION

This section presents a detailed block diagram and logic discussion of each of the system blocks of Paragraph 5.1 and a discussion of the μCPU instruction set and microprogram. The logic discussion makes references to the engineering drawings included in the RX11 and RX8 Print Sets, which are separate documents.

5.2.1 RX8E Interface

Figure 5-7 presents a block diagram of the RX8E interface. The page references in the following discussion are from the RX8 Print Set, which is a separate document.

5.2.1.1 Device Select and IOT Decoder – The Device Select and IOT Decoder Logic, shown on page D2, decodes RX8E instructions from the memory data bus and generates signals to the Interrupt Control and Skip Logic, the C Line Control Logic, and the Sequence and Function Control Logic. Device selection codes are determined by the switch configuration with relation to the state of MD6, MD7, and MD8. When the correct code for the RX8E is input to the Device Select Logic on MD03 L to MD08 L and I/O PAUSE L is asserted, MD09 L to MD11 L are decoded by the 7442 decoder, and signal INTERNAL I/O L is asserted on the Omnibus. I/O PAUSE L is present anytime an IOT instruction is being executed by the program. INTERNAL I/O L prevents the processor from executing other I/O transfers (IOTs) while this instruction is being executed.

The 7442 is a BCD to decimal decoder. All 0s applied to inputs A, B, and C (C is MSB) will cause pin 1, which is unused, to be asserted low. An input of 001 (C is MSB) will cause signal LCD IOT L to be asserted. An input of 010 (decimal 2) will cause XFER IOT L to be asserted. Therefore, for each function code input on MD09 L to MD11 L, only one of the output lines of the 7442 will be asserted. The function codes are further explained in Paragraph 4.4.

5.2.1.2 Interrupt Control and Skip Logic – The Interrupt Control Logic, shown on page D2, asserts the BUS INT RQST L signal on the Omnibus. Bit 11 of the data bus must be set and an INTERRUPT IOT L must be decoded by the RX8E to set the Interrupt Enable flip-flop. The combination of the Interrupt Enable and Buffered Done flip-flops will assert BUS INT RQST L. Setting the Buffered Done flip-flop indicates that no RX01 function is currently in progress.

The Skip Logic implements the three IOT commands Skip on Transfer Request Flag (STR), Skip on Error Flag (SER), and Skip on Done Flag (SDN) as described in Paragraph 4.2.

NOTE

When using these instructions, the respective flags are cleared after they are tested (Paragraph 4.6).

Signal SKIP L will be asserted if any of the above instructions are decoded by the IOT decoder and the respective flag has been asserted by the RX01.

The RX8E asserts the RX8E flags by causing a positive transition on the clock inputs of flip-flops XFER REQ, ERR, and DONE. The signal MAINT (1) L will directly set the Skip flags to allow the Skip IOTs to assert the BUS SKIP L signal when decoded by the IOT decoder.

5.2.1.3 C Line Select Logic – The C Line Select Logic (page D3) controls the direction of data flow between the processor AC and the data bus and determines whether or not the AC is cleared upon completion of the transfer.

C0 L will be asserted during an LCD (Load Command) instruction when signal LCD IOT L is asserted. Assertion of C1 L requires XFER IOT H to be asserted and either MAINT (1) L or B DONE L to be asserted or WRT H to be negated. Data transfers occur under the control of the C bits according to Table 5-1.

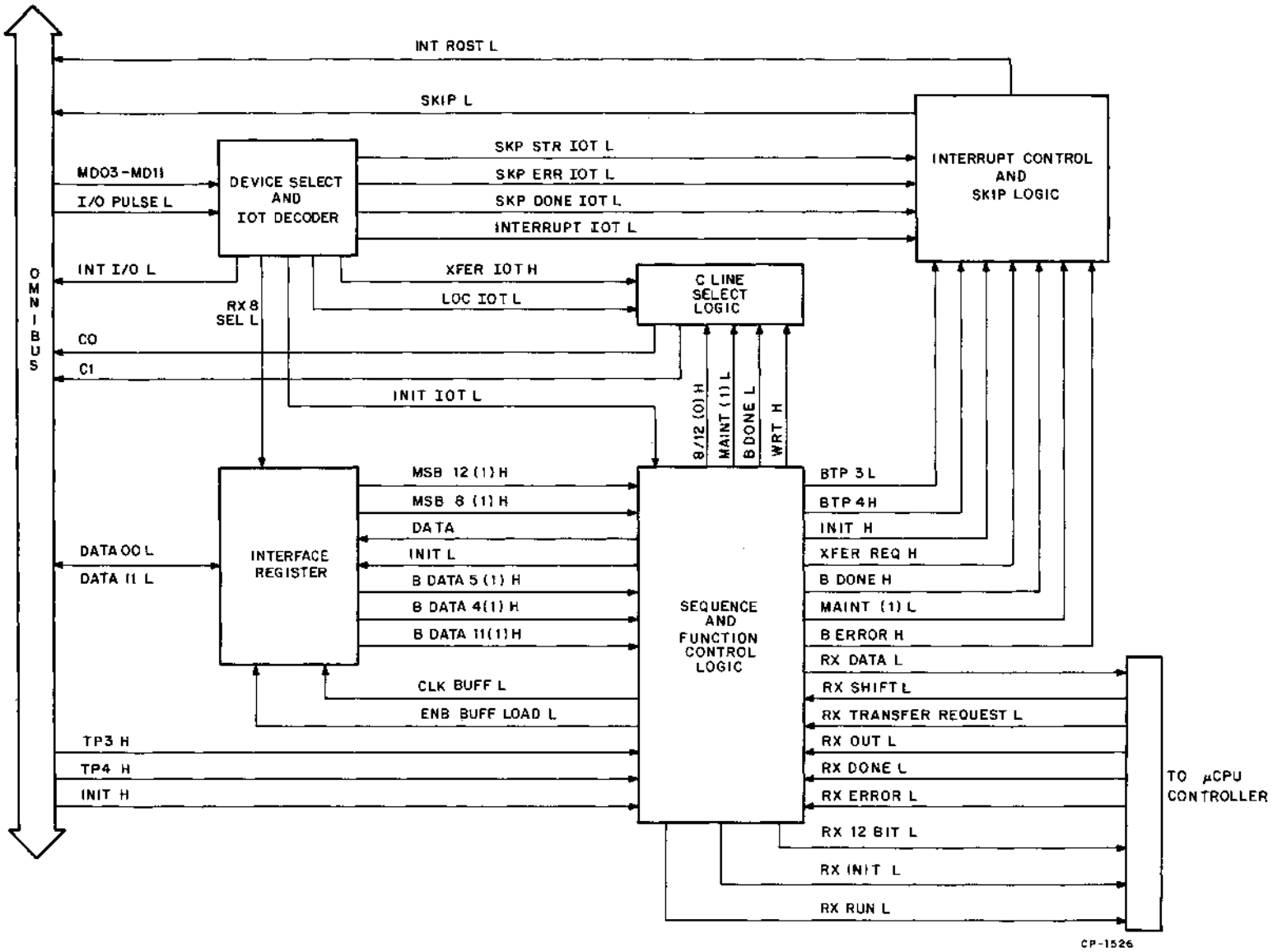


Figure 5-7 RX8E Interface Block Diagram

Table 5-1
C Line Transfer Control Signals

Type of Transfer	C0	C1	Action Required by RX8E Interface	Action by Processor
Output AC to data bus; AC unchanged.	H	H	Load data bus into buffer.	Transfers AC to data bus. AC remains unchanged.
Output AC to data bus; AC cleared.	L	H	Load data bus into buffer. Ground C0.	Transfers AC to data bus and clears AC.
Input AC ORed with peripheral data	H	L	Gate peripheral data to data bus. Ground C1.	AC ORed with peripheral data.
Jam input data bus to AC.	L	L	Gate peripheral data to data bus. Ground C0 and C1.	Transfers data bus to AC register.

5.2.1.4 Interface Register – The Interface register shown on page D3 is made up of three 8271 4-bit shift registers. This register temporarily stores data during transfers from the Omnibus to the RX01 μCPU controller or during transfers from the RX01 μCPU controller to the Omnibus.

During a data transfer from the Omnibus (Fill Buffer), the 12 parallel data lines to the register are enabled by signal RX8 SEL L from the Device Select Logic. Data is parallel loaded into the register when signals ENB BUFF LOAD L and CLK BUFF L are asserted. In shifting data out of the register serially for transmission to the μCPU controller, ENB BUFF LOAD L must be negated. Signal CLK BUFF L from the Sequence and Function Control Logic clocks data out of the buffer (Paragraph 5.2.1.5).

During data transfer to the Omnibus (Empty Buffer), serial data from the μCPU controller is shifted into the buffer. ENB BUFF LOAD L must be negated while CLK BUFF L supplies the clock pulses. The parallel data is enabled from the outputs of the register when MAINT (1) H, RD H, or B DONE H is asserted, and when XFER IOT L is asserted as decoded by the IOT decoder. Only eight bits of data will be output if signal 8/12 (0) H is low; otherwise, 12 bits will be transmitted.

5.2.1.5 Sequence and Function Control Logic – The Sequence and Function Control Logic shown on pages D2 and D3 performs six distinct functions:

1. Controls loading and shifting of the Interface register to and from the μCPU controller.
2. Senses 8- or 12-bit mode and outputs RX 12 BIT L.
3. Senses maintenance conditions.
4. Generates INIT L signal to the μCPU controller.
5. Generates RUN L signal to the μCPU controller.
6. Generates a parity bit for the serial bit stream to the RX01.

Interface register operation is controlled by signals ENB BUFF LOAD L and CLK BUFF L generated by the Sequence and Function Control Logic. To assert ENB BUFF LOAD L, signal RX OUT L cannot be asserted and either RX TRANSFER REQUEST L or RX DONE L must be asserted.

In parallel data entry to the buffer, CLK BUFF L will be asserted if any of the following conditions hold:

1. ENB BUFF LOAD L is asserted.

2. Either LCD or XDR instructions are being performed.

3. Signal BUS TP3 H is asserted.

In serial data entry to the buffer, the CLK BUFF L pulses are derived from the RX SHIFT L pulses from the μCPU controller.

The 8/12 flip-flop will set and signal 8/12 (1) H will be asserted if BUS DATA 5 L is asserted during a Load Command (LCD) operation. Signal 8/12 (1) H is used to control whether 8 or 12 bits of data are transferred to or from the Omnibus and whether 8 or 12 bits of data are transferred between the RX8E and the μCPU controller.

The MAINT flip-flop will set, and signal MAINT (1) H will be asserted if BUS DATA 4 L is asserted during a LCD operation. Signal MAINT (1) H is used to allow parallel writing and reading of the Interface register from the Omnibus. It is also used to assert signal RUN L and C Line Control signals.

An Initialize signal to the μCPU controller (RX INIT L) is generated either by a BUS INIT H signal from the Omnibus or an INIT IOT L decoded from the IOT decoder.

The RUN L signal, which is used to initiate communication between the interface and μCPU controller, is asserted by setting the Run flip-flop. This flip-flop is clocked either in Command Transfer mode when LCD IOT H is asserted or Data Transfer mode to or from the μCPU controller when XFER IOT H is asserted. (RUN L cannot be asserted if DONE L is asserted.) RX BUSY L and INIT L must both be high for a RUN L signal to be asserted. Assertion of either RX BUSY L or INIT L will clear the Run flip-flop.

5.2.2 RX11 Interface

A block diagram of the RX11 interface is shown in Figure 5-8. In the following discussion, it is assumed that the reader is familiar with Unibus operations as described in the *PDP-11 Peripherals and Interfacing Handbook.* Page references are to the RX11 Print Set, which is a separate document.

5.2.2.1 Address Decoder – The address decoder determines whether its associated RX11 is being addressed and whether control information or data is being transferred.

The hardware on page D2 is a combinational logic network that decodes two discrete addresses assigned to the RX11. Address bits (17:13) must always be asserted to satisfy the decoder. The state of address bits A (12:03) is determined by the placement of jumpers A12 through A3 on the board. For each of these bits, one 8242 exclusive-NOR gate is used. Insertion of a jumper for a particular bit position stores a 0 on one leg of the 8242, so that a 1 appearing on the other leg causes the output to go low. This is a mismatch condition which is met when the associated address bit is a 1. When both legs match (1s or 0s on both), the output is high. The output of these 12 gates are wire-ORed and applied to pin 9 of the 7400 NAND gate. Pin 10 of this gate receives the NANDed signal of A (17:13) and BUS MSYN. Pin 8, the output of this gate, is signal REG SELECT L and is asserted when the proper Unibus addresses are decoded.

The states of address bits A02 and A01 and REG SELECT L are used to produce signals D2 SELECT 00 H or D2 SELECT 02 H. The state of A01 determines which of the mutually exclusive signals, D2 SELECT 00 H or D2 SELECT 02 H, is generated. If BUS A01 L is asserted, D2 SELECT 02 H is asserted. If BUS A01 L is negated, D2 SELECT 00 H is asserted. These signals, in turn, allow access to the RXCS register as in the case of D2 SELECT 00 H asserted, or to the RXDB register as in the case of D2 SELECT 02 H asserted. (Refer to Paragraph 3.2 for register descriptions.)

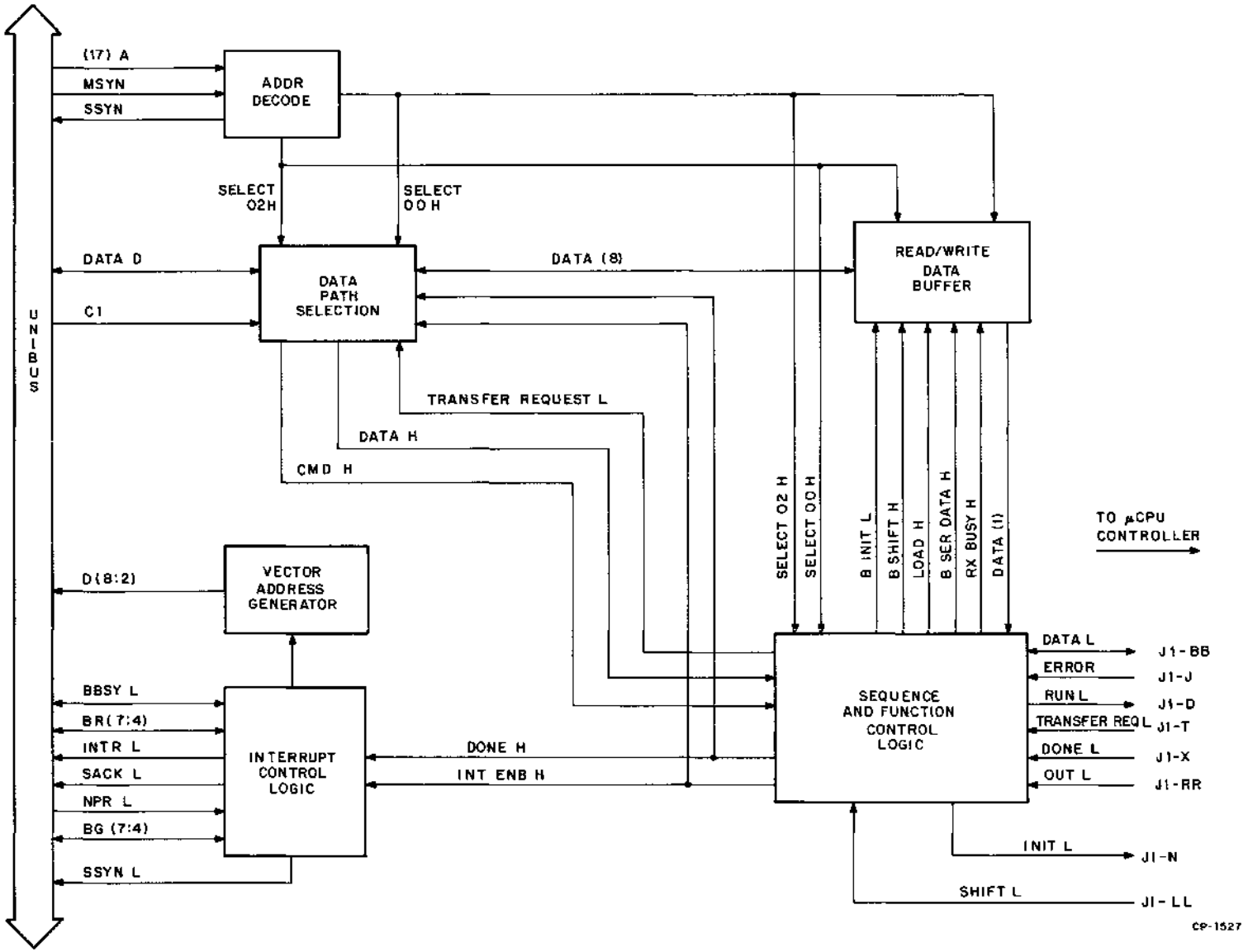


Figure 5-8 RX11 Interface Block Diagram

5.2.2.2 Data Path Selection – Data Path Selection Logic is shown on pages D2 and D3. Signal BUS C1 L from the Unibus controls whether Data Out or Data In operation is to be executed. Assertion of BUS C1 L indicates Data Out (from Unibus), and negation of BUS C1 L indicates Data In (to Unibus). Signals D2 OUT H and its complement D2 IN H from page D2 are derived from BUS C1 L and are input to the rest of the Data Path Selection Logic on page D3. With BUS C1 L negated and D2 SELECT 02 H asserted, all eight bits from the data buffer are enabled through the 8838 bus transceivers. With BUS C1 L negated and D2 SELECT 00 H asserted, only lines BUS D04 through BUS D07, providing control and status information (RXCS), are enabled. With BUS C1 L asserted, none of the 8838s are enabled, and data is being input from the Unibus on lines BUS D00 through BUS D07.

The 74157 multiplexer controls passage of status and control information (RXCS) in the form of signals D3 DONE H, D3 INT ENB (1) H, and D3 TRANSFER REQUEST H from the Sequence and Function Control Logic or passage of data from the Read/Write Data Buffer register (RXDB). If D2 SELECT 02 H is asserted, then data is output from the 74157. If D2 SELECT 02 H is negated, then control information (RXCS) is output from the 74157.

5.2.2.3 Interface Register – The Interface register is a 74199 eight-bit, parallel load, shift register shown on page D3. Data transfer through the register can take place from the Unibus to the μCPU controller or from the μCPU controller to the Unibus.

In data transfer to the RX11 from the Unibus, parallel data is loaded into the register when D3 RX BUSY H is negated and D3 LOAD H is asserted. Data is serially shifted in the register from Q_A to Q_H by the D3 B SHIFT H signal derived from the μCPU controller when D3 RX BUSY H is asserted. Serial data is shifted to the Sequence and Function Control Logic, which transmits data to the μCPU controller (Paragraph 5.2.2.4).

Data is assembled in a serial fashion for parallel transfer on the Unibus. Serial data is input at D3 B SER DATA H and shifted by D3 B SHIFT H when D3 RX BUSY H is asserted and D3 LOAD H is negated. The eight bits of parallel data appearing at the output of the buffer are input to the Data Path Selection Logic for transmission to the Unibus.

5.2.2.4 Sequence and Function Control Logic – The Sequence and Function Control Logic schematics are shown on page D3 and in the lower left-hand corner of page D2. This portion of the RX11 interface provides key signals to control the Interface register and the Interrupt Control Logic as shown in Figure 5-8. Operation of this circuitry is controlled by signals from the μCPU controller and D2 SELECT 00 H and D2 SELECT 02 H from the address decoder.

Signals RX TRANSFER REQUEST L, RX OUT L, RX DONE L, and RX RUN L control data transfer between the interface and the μCPU controller. The RX RUN L signal from the RX11 interface initiates communication between the RX11 interface and the μCPU controller. The Run flip-flop is set in passing either a command from interface to controller or data between interface and controller. Run asserted while Done is true passes a command from interface to controller. Run asserted while Done is false signals transfer of data to or from the controller.

Once a particular function has been decoded by the μCPU controller, it requests a data transfer by assertion of RX TRANSFER REQUEST L. The access of the RXDB in the RX11 interface sets the Run flip-flop and thereby asserts RX RUN L. The Run flip-flop is cleared either by assertion of D2 B INIT H or RX BUSY H. RX BUSY H is asserted when both RX TRANSFER REQUEST L and RX DONE L are negated. Assertion of RX BUSY H also allows the Interface register to shift serially in communicating between μCPU controller and interface.

RX OUT L from the μCPU controller determines in which direction the data transfer is about to take place. When RX OUT L is asserted, the direction of data transfer is from controller to interface. When RX OUT L is negated, the direction of data transfer is from interface to controller.

On transfers from controller to interface, assertion of RX TRANSFER REQUEST L indicates that the next data element has been assembled in the RXDB. Transfer of the next data element is initiated by RX RUN L. On transfers from interface to controller, assertion of RX TRANSFER REQUEST L indicates that the controller is prepared to accept the next element of data. Arrival of the next data element will be signaled by assertion of RX RUN L.

The three signals D3 DONE II, D3 INT ENB (1) H, and D3 TRANSFER REQUEST H from the Sequence and Function Control Logic to the Data Path Selection Logic represent the three bits that may be read in the Control and Status (RXCS) register. A functional programming description of this register is given in Paragraph 3.3.

During serial data transfer from the RXDB, a 8281 binary counter and a 74H106 JK flip-flop are used to count eight bits of data and to append the parity bit to the data element.

An error indication from the μCPU controller results in assertion of RX ERROR L. This indication is passed to the Unibus when a read from the RXCS to the Unibus is performed. When this occurs, signal BUS D15 L is asserted.

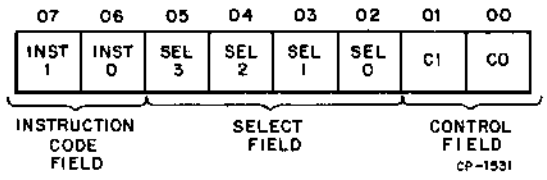
5.2.2.5 Interrupt Control Logic – The Interrupt Control Logic, shown on page D2, is a combinational logic network that receives and generates the control signals required for the RX11 to become bus master. With signals D3 DONE H and D3 INT ENB H both asserted, D2 BUS REQUEST L will be generated if the SACK and BBSY flip-flops are not set. The D2 BUS REQUEST L signal is routed to the appropriate bus request line (normally BR5) through the priority plug shown on page D3. Etch on the plug selects both request and grant lines. When the bus grant signal is generated by the processor, it is routed via the priority plug and becomes signal D3 BG IN H. This signal clocks the GRANT flip-flop and the SACK flip-flop. The SACK flip-flop is set because the RX11 had requested bus mastership. The SACK flip-flop is cleared and the BBSY flip-flop set when BUS BBSY L, BUS SSYN L, and D3 BG IN H are negated on the Unibus. Thus the BUS BBSY L signal will be asserted again by the RX11, which is now bus master. The BBSY signal is inverted and applied to the vector address generator, generating the BUS INTR L signal and the vector address of 264.

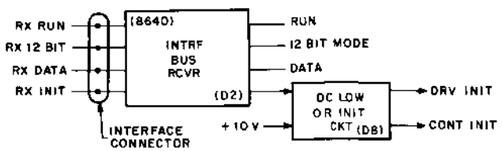
5.2.2.6 Vector Address Generator – The vector address generator, shown on page D2, consists of eight bus drivers that are used to generate a vector address and the BUS INTR L signals. When BUS BBSY L is asserted by the RX11, the inputs to the bus drivers are active. Seven drivers are connected to the Unibus data lines D (08:02) via jumpers. The placement of these jumpers determines the vector address.

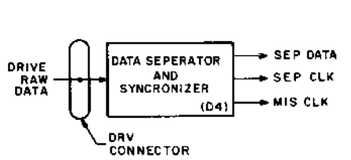
5.2.3 Microprogrammed Controller (μCPU) Hardware – A block diagram of the μCPU controller is shown in Figure 5-9. The controller is a hardware microprocessor controlled by a ROM with 1536 eight-bit words; the ROM contains the microprogram. This section discusses the various parts of the hardware while Paragraph 5.2.5 discusses the microprogram.

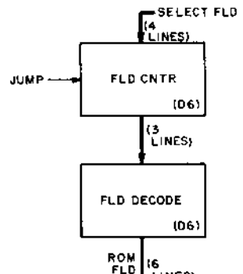
5.2.3.1 Control ROM and Memory Buffer – The heart of the controller is the ROM, which contains the microprogram shown on page D6. All control and processing activities executed by the RX01 are controlled by the microcode sequences stored within the ROM. The ROM is divided into six fields, each with a storage capacity of 256 eight-bit words. Sequencing through the ROM microcode is accomplished by updating the contents of the program counter (PC) every 200 ns. The eight bits from the PC indicate the address of the next instruction to be executed.

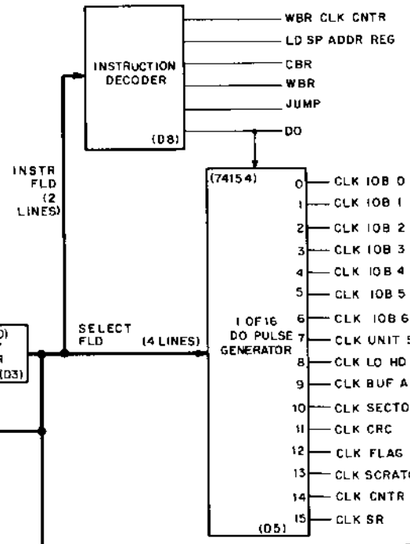
The eight-bit instruction addressed by the PC is loaded into the memory buffer on assertion of LD MB + CLK PC L. Each instruction consists of three fields that are sampled by the μCPU logic circuits to allow processing action appropriate to the instruction to be taken. The three fields can be defined as follows:

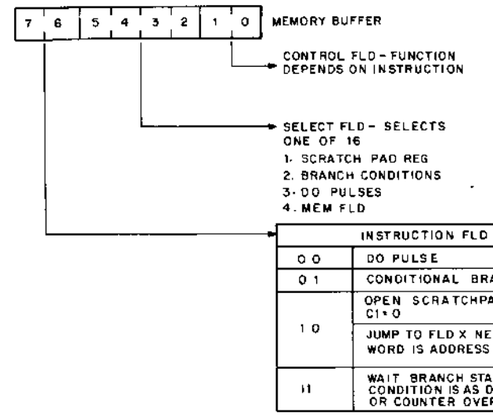




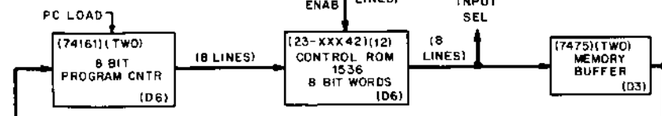


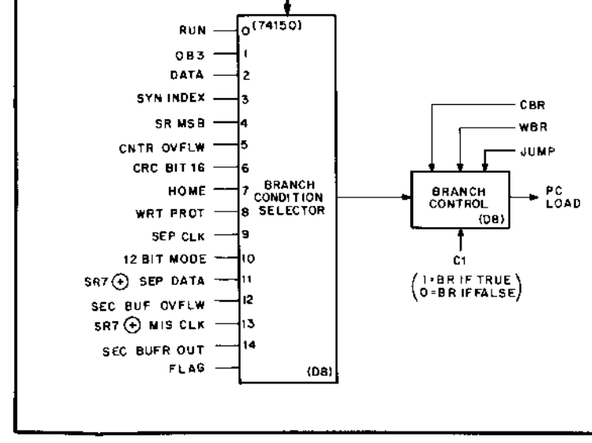


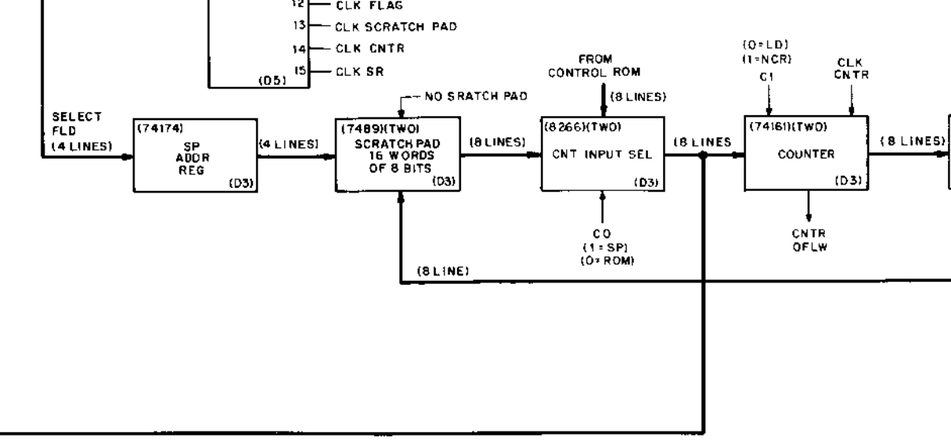


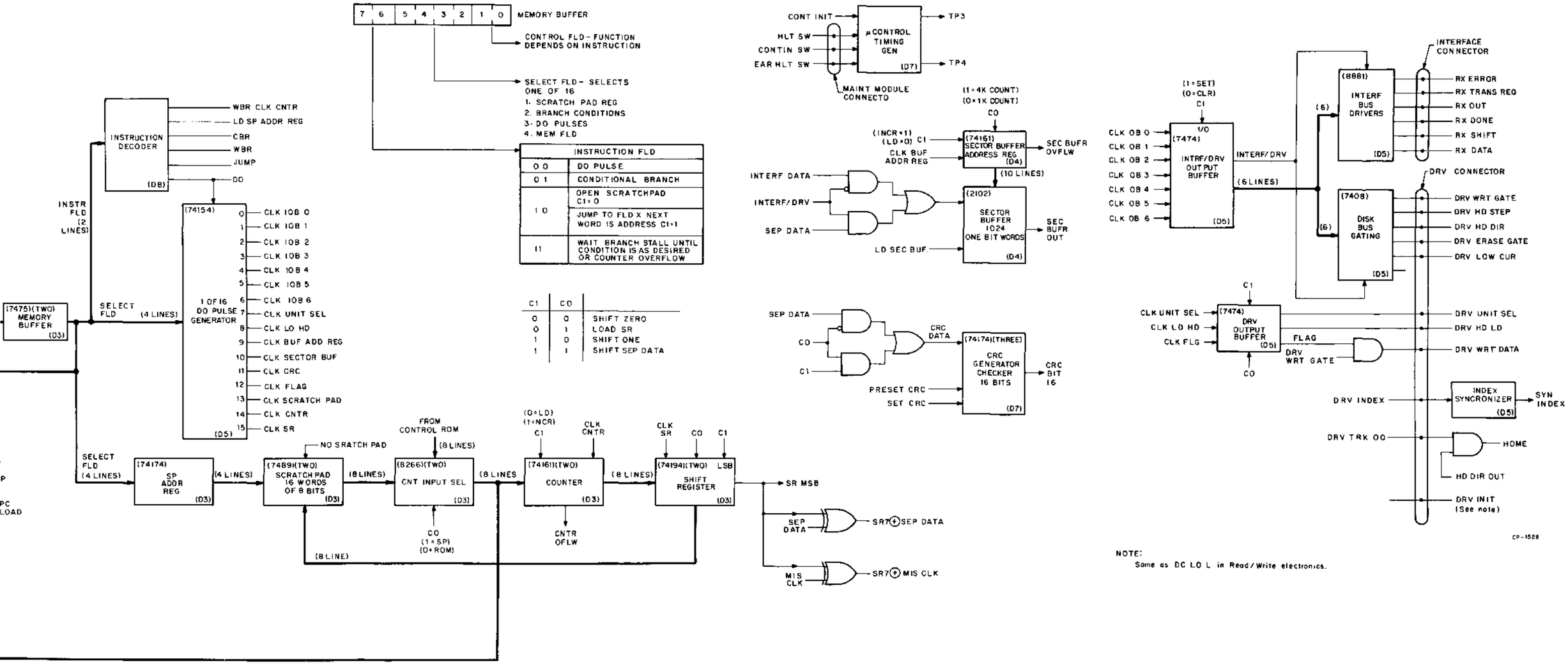


C1	C0	
0	0	SHIFT ZERO
0	1	LOAD SR
1	0	SHIFT ONE
1	1	SHIFT SEP DAT









NOTE:
Same as DC LO L in Read/Write electronics.

Figure 5-9 μCPU Controller Block Diagram

5.2.3.4 Do Pulse Generator – Detection of a Do instruction by the instruction decoder and assertion of signal TP3 L will enable the Do pulse generator (page D5), which is a 74154 decoder. The function select bits determine which of the outputs is asserted. All outputs supply clocking to various flip-flops, counters, and shift registers, depending on the function select bits. If the select field is 0000, signal CLK IOB 0 is asserted; if the select field is 0001, signal CLK IOB 1 is asserted; etc.

5.2.3.5 Branch Condition Selector and Control – The 74150 branch condition selector (page D8) is always enabled during a Branch instruction. The function select field determines which data line is selected for input to the Branch Control Logic. The output signals from the Branch Control Logic, PC LOAD EN L and LD MB + CLK PC L, are used to either increment the program counter or load in a new address. LD MB + CLK PC L is also used to load the memory buffer.

The C1 bit, which the firmware has converted to data, is compared with the output of the branch condition selector. If the conditions match, the PC LOAD EN L and LD MB + CLK PC L signals are enabled. The WBR CLK CNTR L signal is asserted when a WBR instruction is decoded by the instruction decoder.

5.2.3.6 Scratch Pad Address Register and Scratch Pad – During execution of an Open Scratch Pad instruction, the function select field contains an address in the Scratch Pad. The four-bit function select field is input to the Scratch Pad Address register (page D3), which consists of four D-type flip-flops. The function select field appears at the output of this register upon assertion of the CLK SP ADDR REG L signal generated by the instruction decoder.

The Scratch Pad (page D3) itself consists of two 7489s (64-bit read/write memory), capable of storing 16 eight-bit words. Data from the shift register is written into the address indicated by the Scratch Pad Address register when LD SCRATCH PAD L is asserted. Reading occurs when this signal is negated. Data read out is input to the Counter Input Selector Logic.

5.2.3.7 Counter Input Selector, Counter, and Shift Register – The counter input selector, counter, and shift register (page D3) act as buffering circuitry for information flow in and out of the Scratch Pad. Signals C0 and C1 control operation of this logic.

The counter input selector consists of two 8266 multiplexers that select eight parallel bits from either the ROM or the Scratch Pad, depending on the state of C0(1) H. If C0(1) H is asserted, the Scratch Pad data is selected; if C0(1) H is negated, the ROM data is selected. The state of C0 is determined by the low order bit of the instruction code from the memory buffer.

The outputs of the counter input selector are presented to both the program counter and the counter. During execution of a Jump instruction and Branch instruction when the branch conditions are met, a new address is loaded into the program counter. The control field of the instruction will determine the source of the new address. If the source is the ROM, the address is in the location following the instruction. If the source is the Scratch Pad, some previous calculation was performed to determine the new address.

The counter is made up of two 74161 synchronous four-bit counters. The output of the counter input selector is loaded in the counter if signal C1(1) H is negated. If C1(1) H is asserted, the counter increments upon detection of signal CLK CNTR L. During execution of a DO instruction, the Do pulse generator supplies the clocking signal if the select field is correct. An overflow condition, indicated by all 1s in the counter, is detected by assertion of signal CNTR OVFLW H, which is input to the branch condition selector.

Data output from the Scratch Pad counter is loaded into the shift regsiter if C0(1) L is asserted and C1(1) H is negated. Data appears at the outputs after the positive transition of the clock input CLK SR L. When C0(1) L and C1(1) H are asserted, data from the data separator is serially shifted into the shift register by signal CLK SR L. Therefore, input to the Scratch Pad is either by way of serial data from the data separator or parallel data from the counter. The data from the data separator originates from the read/write electronics during access from the diskette. SRMSB, the MSB of the shift register, is exclusive-ORed with data [SEP DATA (1) L] and a missing clock indication [MIS CLK (1) L] from the data separator for input to the branch condition selector. SRMSB (1) H alone is also presented to the branch condition selector.

Bits can be shifted into the shift register via CLK SR L, C0, and C1. Signal SR LOAD H is asserted when C0(1) L is asserted and C1(1) H is negated, allowing the shift register to be parallel loaded from the counter. With C0(1) H and C1(1) H both asserted, SEP DATA from the data synchronizer and separator is serially input to the shift register. If C0(1) L is negated, 1s and 0s from the C1(1) H bit stream are loaded into the shift register. In summary, C0 determines shift or load of the shift register; C1 and SEP DATA are two serial bit streams.

C1	C0	
0	0	Shift Zero
0	1	Load Shift Register
1	0	Shift One
1	1	Shift SEP DATA

5.2.3.8 μCPU Timing Generator – The μCPU timing generator, shown on page D7, produces signals TP3 and TP4, which are used as timing signals for the rest of the μCPU controller. The 74H74 flip-flops are used as a wraparound shift register to derive TP3 and TP4 from the 20 MHz oscillator. TP3 and TP4 are 5 MHz signals with a pulse width of 50 ns. In normal operation, a recirculating 1 bit in this shift register causes TP3 to occur before TP4. Inputs to the maintenance module connector and signal INIT + PC L control operation of this shift register.

5.2.3.9 Sector Buffer and Address Register – The 2102 sector buffer, which is a 1024-bit MOS RAM, and the address register, composed of three 74161 synchronous four-bit counters, is shown on page D4. The address register is used as a loop counter for the microprogram as well as an address register for the sector buffer. The ten LSB inputs to the register are grounded, and the upper two MSB inputs are connected to signal C0(1) L.

When used as a loop counter or an address register, C1(1) H is negated and the ten LSB bits of the register are zero. With C1(1) H asserted, the count is incremented by signal CLK BAR L, which occurs once per disk data bit. If C0(1) L is asserted, the two MSB bits are also zero, enabling the counter to count 4096 clocks before the SEC BUF OVFLW H signal is asserted. If C0(1) L is negated, the two MSB bits are one, enabling the counter to count only 1024 clocks before the SEC BUF OVFLW H signal is asserted.

Reading or writing is controlled by a flip-flop. If C0(1) H is asserted and CLK SEC BUF L is asserted, the write enable line to the 2102 is asserted by setting the flip-flop. The CLK SEC BUF L signal is asserted twice per bit, once to set the flip-flop and once again to clear it. It is cleared when C0(1) H is negated and the second CLK SEC BUF L pulse is asserted. Data to be written in the sector buffer is contained in signal SEP DATA (1) L from the read/write electronics while reading from the diskette; the data is contained in signal DATA I L from the interface while writing on the diskette.

In reading data out of the sector buffer, the write enable line is negated, and the serial data stream appears at SEC BUF OUT (1) H as the Buffer Address register is incremented.

5.2.3.10 CRC Generator and Checker – Each sector has a two-byte CRC character for the header field and another two-byte CRC character for the data field (Figure 1-11). The CRC generator and checker shown on page D7 produces the CRC character to be written on the diskette and checks the CRC character read from the diskette. A 16-bit shift register with properly placed exclusive-OR gates implements the polynomial divide algorithm. The CLK CRC L signal clocks the shift register so that the entire header field or data field is divided by a selected CRC divisor which is $2^{15} + 2^{12} + 2^5 + 1$. The mathematical expression for this operation is:

$$\frac{a_n 2^n + \ldots + a_3 2^3 + a_2 2^2 + a_1 2^1 + a_0 2^0}{2^{15} + 2^{12} + 2^5 + 1}$$

where a = coefficient of the bit position

n = number of bit positions in the data block

In writing data on the diskette, each bit is shifted through the CRC generator. This data stream appears on signal C1(1) L and is produced by the firmware. Signal C0(1) L negated will enable this data stream to the CRC register. After the data field has been written, the CRC register contains the remainder of the division, which is the two-byte CRC character that is written after the data field.

During a read operation, the data stream from the data synchronizer and separator, SEP DATA (1) L, is enabled to the CRC register. This data stream is manipulated in the same way as in the write operation. When the CRC character on the diskette is encountered, it is shifted into the CRC generator as if it is part of the data stream. If all the data and CRC bits that were previously written on the diskette are retrieved, the CRC register, which contains the remainder of the division, should be zero. The contents of this register are input to the condition selector, where the firmware checks to see that the register contents are all zero.

5.2.3.11 Data Synchronizer and Separator – The data synchronizer and separator (page D4) separates clocking information from data, identifies missing clocks, and synchronizes the clock to the data. Clocking and data are mixed in the output data stream (Paragraph 1.3.2). In the read/write electronics, one-shots set to produce 200 ns pulses for each transition convert the flux reversal signals to a series of pulses as shown in Figure 5-10. The clock pulses can be separated from the 1 data pulses as shown. If no data pulse occurs between two clock pulses, a 0 bit is indicated. Notice that there is a clock pulse every 4 μs.

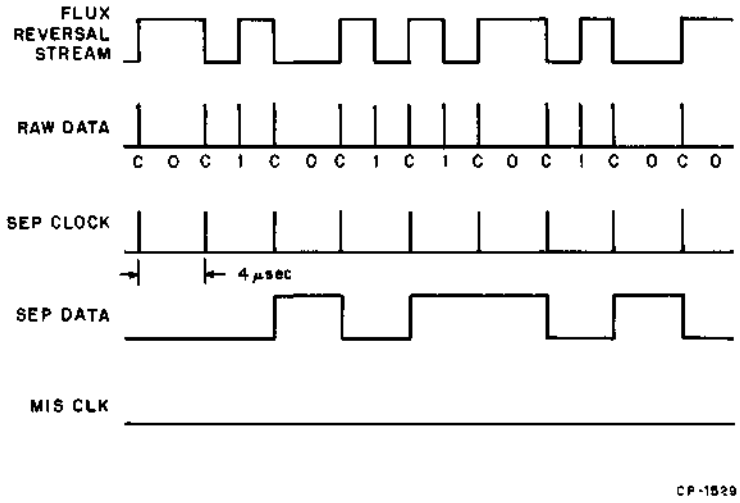


Figure 5-10 Data and Clock Separation

The data synchronizer and separator produces three outputs: separated clock (SEP CLK), separated data (SEP DATA), and missing clock (MIS CLK). SEP DATA is one bit position delayed from the flux reversal stream. In the example shown in Figure 5-10, there are no missing clock indications because the RAW DATA stream does have a clock pulse every 4 μs. The MIS CLK indication is used in locating the ID address mark of the sector header field and the data or deleted data mark of the sector data field. (See Figure 1-9.) Each of these data marks is a unique sequence of data and missing clocks that the microprogram identifies by examining the data synchronizer and separator circuit.

Figure 5-11 is the ID address mark which contains missing clocks. Since clocks must occur every 4 μs, the missing clocks can be identified as $\overline{C}$. SEP CLK output will include clock pulses that are separated by 6 μs and will be out of synchronization with the real clock pulses. SEP DATA will indicate a 1 bit when a data pulse exists between SEP CLK pulses; it will indicate a 0 where there is no data pulse. As in the first case, SEP DATA is delayed by one clock period. MIS CLK is also delayed by one clock period and occurs when SEP CLK pulses occur more than 5 μs apart.

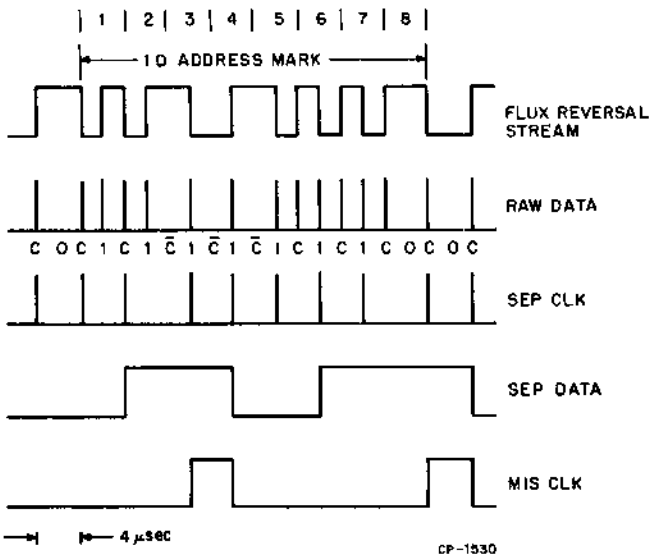


Figure 5-11 ID Address Mark Data Separation

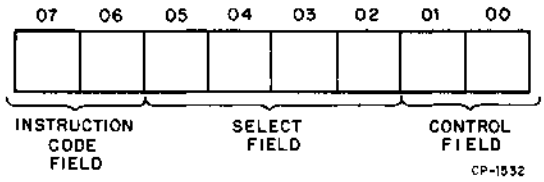
The data synchronizer and separator uses two timing windows to separate data from clocks. If a pulse occurs within 3 μs from a valid clock pulse, it is a data 1 bit. If it occurs between 3 and 5 μs from a valid clock pulse, it is the next clock pulse. If it occurs after 5 μs, there was a MIS CLK. The logic is shown on page D4. The timing windows are produced by two pairs of 74193 counters, which are preset to a given count and clocked by the 20 MHz clock after it has been divided down to 10 MHz. The carry of these counters is used to clear flip-flops to provide timing indications. The END WIND L signal from the 3 μs timer clears the 74H103 flip-flop. The two following 74S74s synchronize the data to the 20 MHz clock. The SEP DATA output delayed by one SEP CLK clock period appears at the output of the 74H103. MIS CLK is produced when the 5 μs timer issues a carry to clear the 74H103. SEP CLK is produced by the 74H103 flip-flop, which is cleared by a state of the 3 μs timer. The following flip-flop synchronizes the SEP CLK signal with BTP3H.

5.2.3.12 Output Circuit – The output circuit shown on page D5 consists of the interface/drive output buffer, drive output buffer, interface bus drivers, disk bus gating, and index synchronizer. Signals CLK IOB0 to CLK IOB6 from the Do pulse generator are used to clock the flip-flops in the interface/drive output buffer. The interface bus drivers and the disk bus gating outputs are appropriate combinations of signals from the interface/drive output buffer that are output to the interface or to the read/write electronics. IOB0 selects the output bus to which the rest of the IOB signals are to be assembled. A further explanation of these signals is given in Paragraphs 5.1.3 and 5.1.4.

The drive output buffer consists of three flip-flops, FLAG, UNIT, and HD LD, which respectively produce signals DRV WRT DATA, DRV SEL 1 H, and DRV HD LD H to the read/write electronics.

The index synchronizer consists of two flip-flops used to synchronize the SEL INDEX H signal from the read/write electronics. The flip-flops are cleared and SYN INDEX (1) H is negated by the Do pulse generator. SYN INDEX (1) H will be asserted by the first TP3 after the SEL INDEX H assertion and input to the branch condition selector.

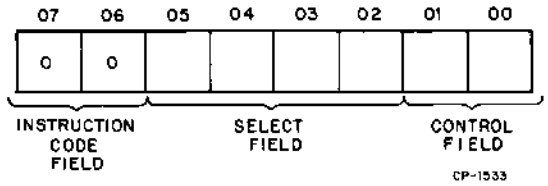
5.2.4 Microprogram Instruction Repertoire – The firmware within the RX01 Read-Only Memory (ROM) employs five different instruction types to implement the various control sequences used to process each function code. Regardless of type, an instruction is made up of eight bits and contains three fields as shown below:



The RX01 μCPU controller distinguishes between the different instruction types by the content of the instruction code field. The select field is used to define subfunctions of a single instruction type. This field is also used for addressing locations in the μCPU Scratch Pad memory. The control field allows for still further definition of the instruction purpose and, in one case, serves to distinguish between two instructions having the same instruction field code. The five basic instructions executable by the μCPU controller are:

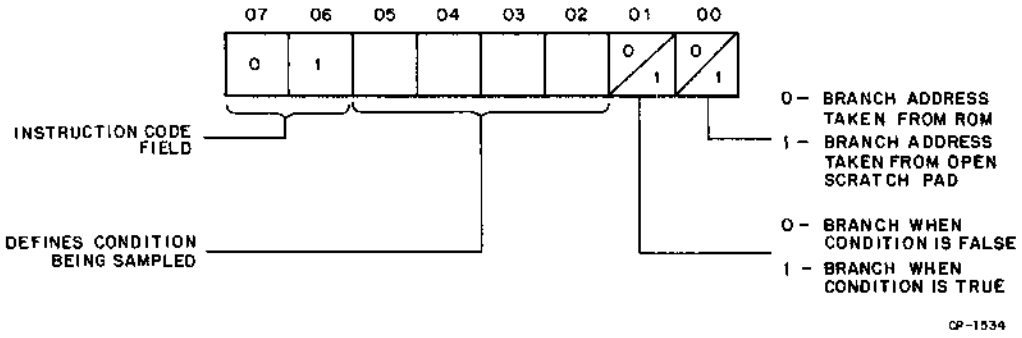
1. DO instruction
2. Conditional Branch
3. Wait Branch
4. Open Scratch Pad
5. Jump

5.2.4.1 DO Instruction – The most frequently executed μCPU firmware instruction is the DO instruction. Its format is:



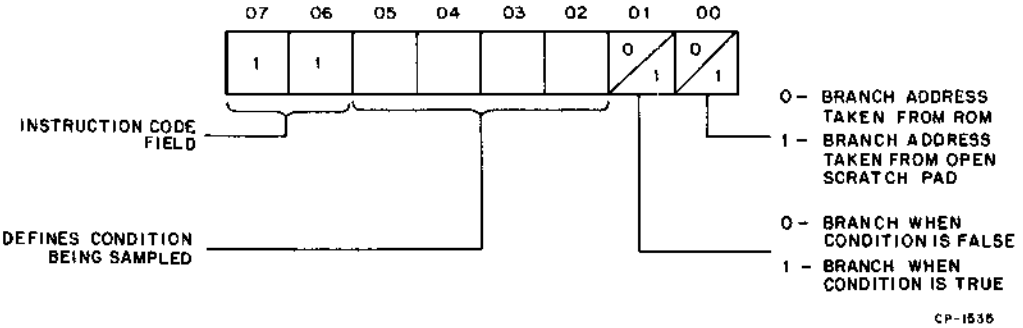
Through use of different function select codes, the DO instruction is used to assert/negate many of the interface lines going to both the interface and the read/write electronics. The DO instruction is also used in shifting data bits to/from the interface for the Empty/Fill Buffer function and writing data bits onto the disk for the Write Sector function. Furthermore, the DO instruction is used for function decoding, parity checking, CRC field generation/detection, and numerous housekeeping functions inherent in the various μCPU controller sequences. A complete breakdown of all DO instruction subfunctions is given in the RX8/RX11 Print Sets.

5.2.4.2 **Conditional Branch** – The Conditional Branch instruction is used to sample status conditions within the RX01. On detection of a given condition, a branch to another area of the ROM occurs. The format of the Conditional Branch instruction is given below:



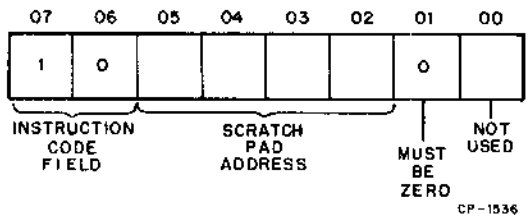
Many conditions are sampled by the Conditional Branch instruction; examples are 12-bit mode, drive ready, read/write head located at track zero, and two index pulses have occurred. A complete breakdown of all Conditional Branch subfunctions is given in the RX8/RX11 Print Sets.

5.2.4.3 **Wait Branch** – The Wait Branch instruction is similar to the Conditional Branch instruction; the difference is that the Wait Branch instruction is used to stall μCPU controller operations until a given condition becomes true. The format of the Wait Branch instruction is given below:



A breakdown of all Wait Branch subfunctions is given in the RX8/RX11 Print Sets.

5.2.4.4 **Open Scratch Pad** – The Open Scratch Pad instruction is used to address (select) any of 16 locations in the μCPU controller prior to executing a read/write operation. The format of the Open Scratch Pad instruction is given below:



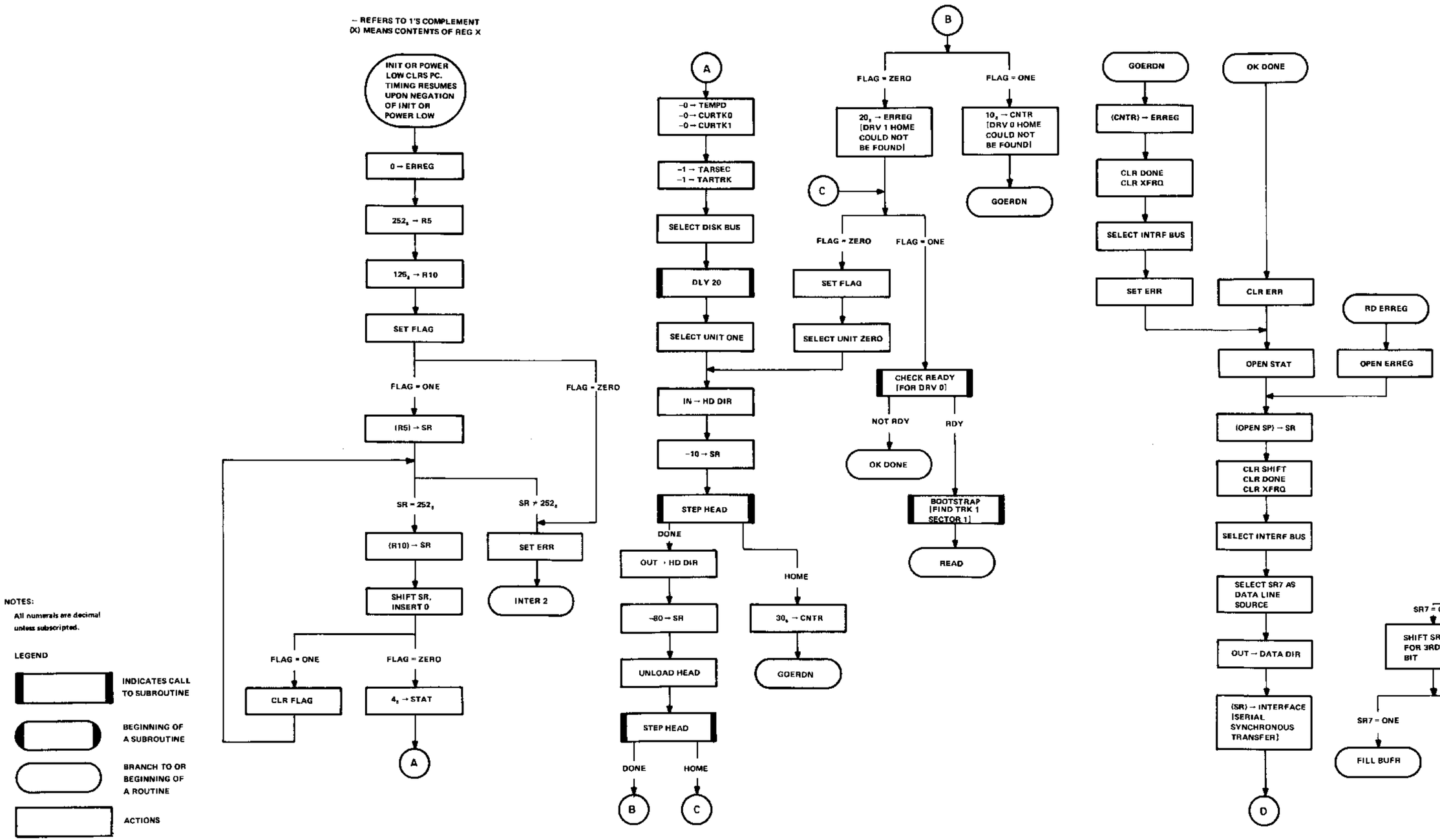
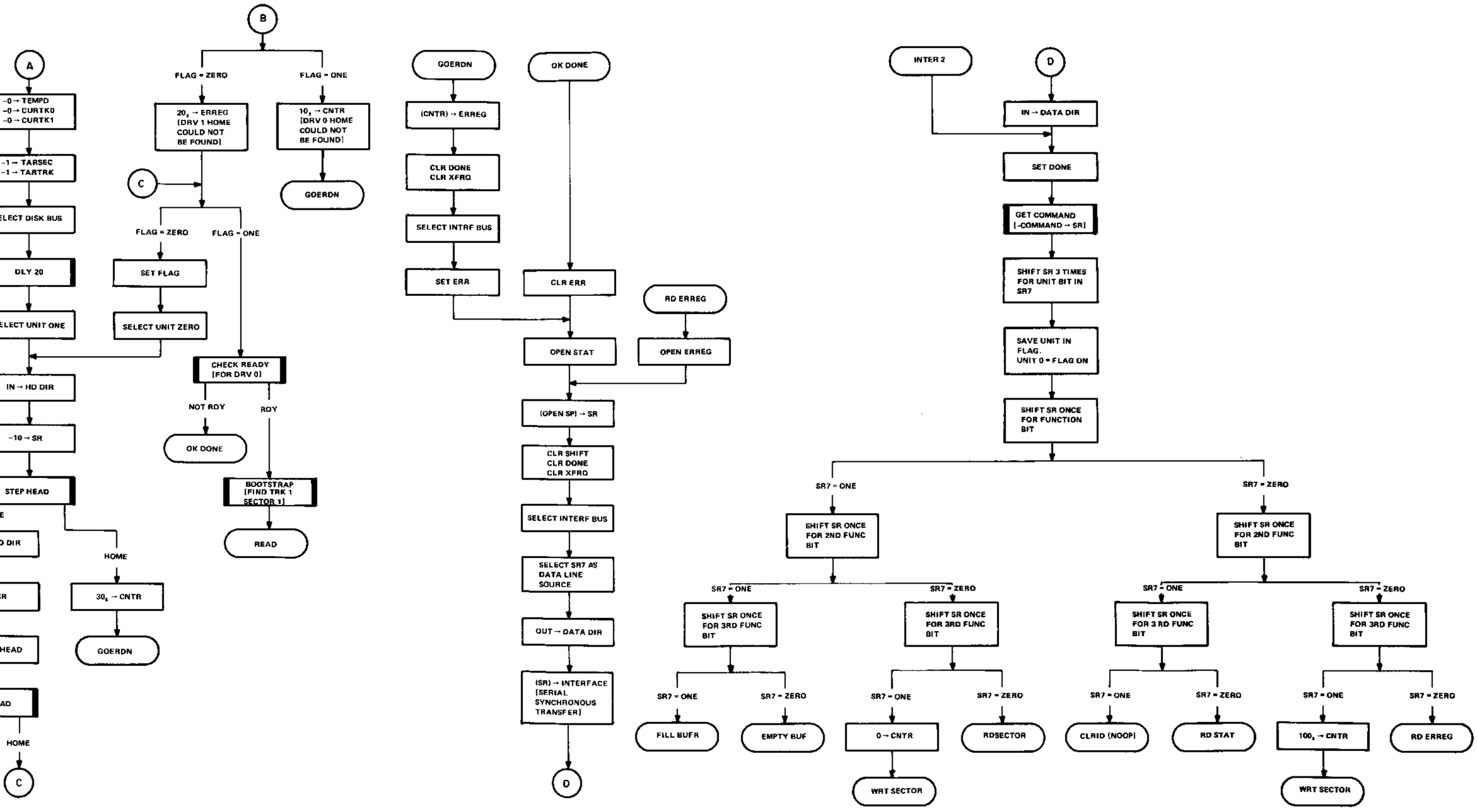


Figure 5-12 Initialize and Function Decode Flowchart


A
-0→ TEMPD
-0→ CURTK0
-0→ CURTK1
-1→ TARSEC
-1→ TARTRK
ELECT DISK BUS
DLY 20
ELECT UNIT ONE
IN→ HD DIR
-10→ SR
STEP HEAD
HOME
30₈ → CNTR
GOERDN
B
FLAG = ZERO
FLAG = ONE
20₈ → ERREG
[DRV 1 HOME
COULD NOT
BE FOUND]
10₈ → CNTR
[DRV 0 HOME
COULD NOT
BE FOUND]
GOERDN
C
FLAG = ZERO
FLAG = ONE
SET FLAG
SELECT UNIT ZERO
CHECK READY
[FOR DRV 0]
NOT RDY
RDY
OK DONE
BOOTSTRAP
[FIND TRK 1
SECTOR 1]
READ
GOERDN
(CNTR) → ERREG
CLR DONE
CLR XFRQ
SELECT INTRF BUS
SET ERR
OK DONE
CLR ERR
RD ERREG
OPEN STAT
OPEN ERREG
(OPEN SP) → SR
CLR SHIFT
CLR DONE
CLR XFRQ
SELECT INTERF BUS
SELECT SR7 AS
DATA LINE
SOURCE
OUT→ DATA DIR
(SR) → INTERFACE
[SERIAL
SYNCHRONOUS
TRANSFER]
D
INTER 2
D
IN→ DATA DIR
SET DONE
GET COMMAND
[-COMMAND → SR]
SHIFT SR 3 TIMES
FOR UNIT BIT IN
SR7
SAVE UNIT IN
FLAG.
UNIT 0 = FLAG ON
SHIFT SR ONCE
FOR FUNCTION
BIT
SR7 = ONE
SR7 = ZERO
SHIFT SR ONCE
FOR 2ND FUNC
BIT
SHIFT SR ONCE
FOR 2ND FUNC
BIT
SR7 = ONE
SR7 = ZERO
SR7 = ONE
SR7 = ZERO
SHIFT SR ONCE
FOR 3RD FUNC
BIT
SHIFT SR ONCE
FOR 3RD FUNC
BIT
SHIFT SR ONCE
FOR 3 RD FUNC
BIT
SHIFT SR ONCE
FOR 3RD FUNC
BIT
SR7 = ONE
SR7 = ZERO
SR7 = ONE
SR7 = ZERO
SR7 = ONE
SR7 = ZERO
SR7 = ONE
SR7 = ZERO
FILL BUFR
EMPTY BUF
0→ CNTR
RDSECTOR
CLRID (NOOP)
RD STAT
100₈ → CNTR
RD ERREG
WRT SECTOR
WRT SECTOR

CP-1544

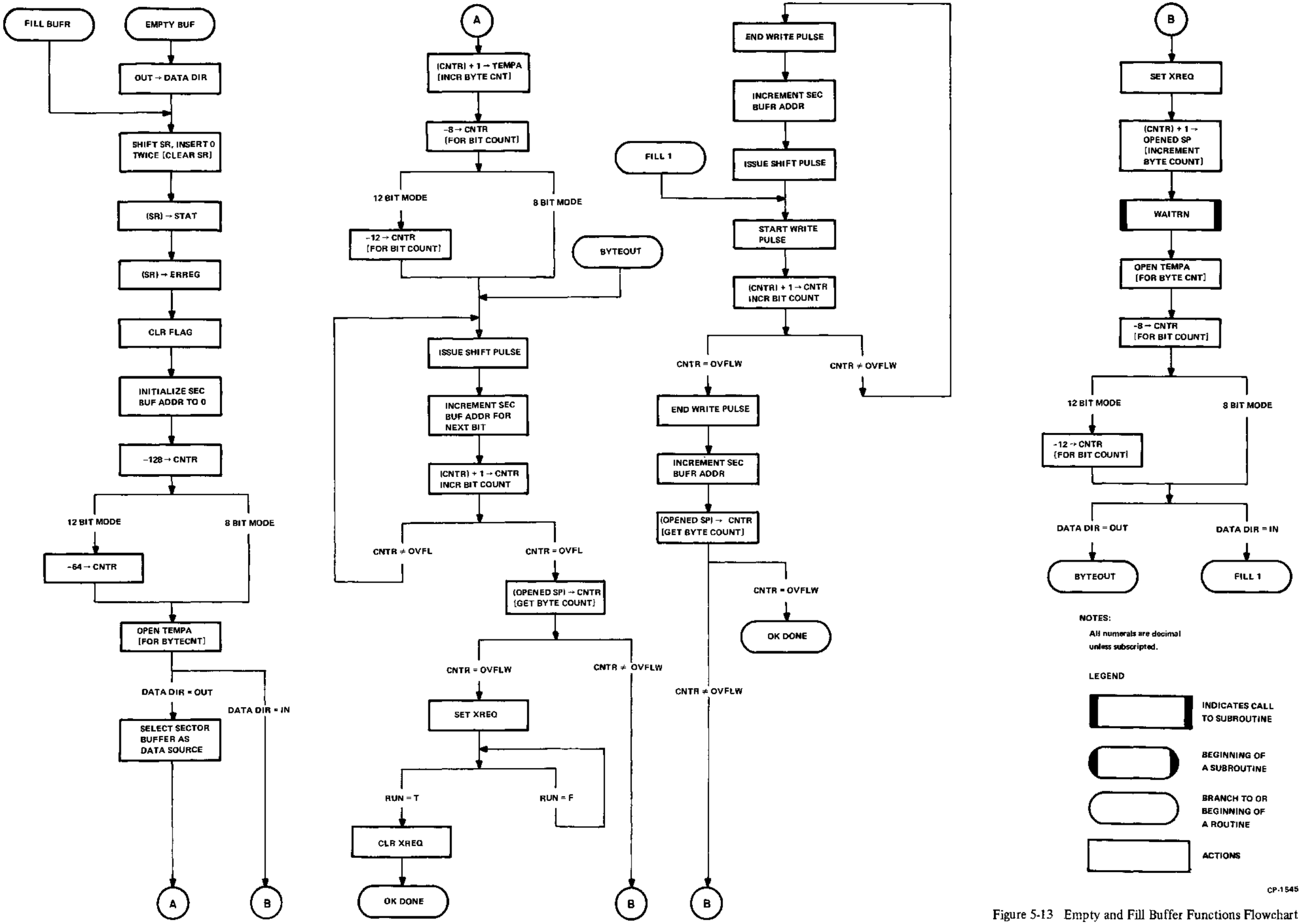


Figure 5-13 Empty and Fill Buffer Functions Flowchart

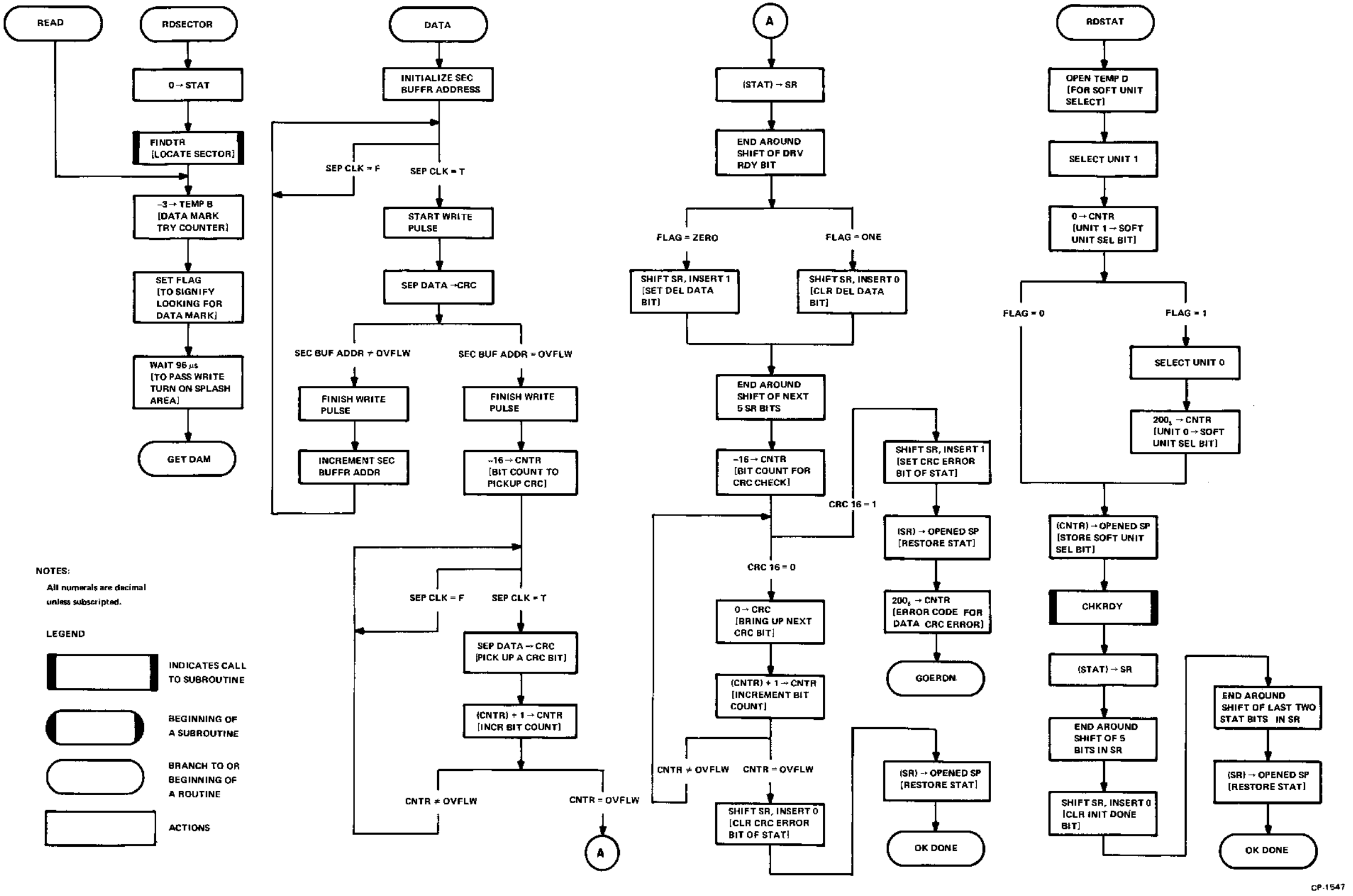


Figure 5-14 Read Sector and Read Status Functions Flowchart

The Read Status function (Figure 5-14) checks to see if the Drive Ready conditions are met by calling the CHKRDY subroutine. If they are, the Drive Ready bit of the Status register is set, and the system status is checked. If not, the Drive Ready bit is not set, and the system status is checked by a branch to the DONE condition.

The functions Write Sector and Write Deleted Data Sector (Figure 5-15) are similar except that the Deleted Data flag is set in the Write Deleted Data Sector function. The diskette address and sector are found and WRITE GATE and ERASE GATE signals are sent to the read/write electronics. The sector must be correctly formatted with proper header, header CRC, 1024 bits of data, and data CRC. After the WRITE GATE and ERASE GATE signals are negated and the next header is located, a branch is made back to the DONE condition to check status.

The FINDTR subroutine (Figure 5-16) locates the track as specified by the diskette address. Status and Error Scratch Pad locations are cleared. The Drive Select bit is interrogated to determine which drive is being used. The sector address is moved from the interface to the controller shift register by the subroutine GETWRD, and its parity is checked. Then the eight-bit track address is moved from the interface to the target track register, and parity and track legality are checked. The current track address of the drive selected is compared with the target track address by the subroutine MAGCOM to determine if the head must step in or out to reach the target track. Head stepping is accomplished by the subroutine STEPHD once the proper track is located and the head is loaded. If the track address is greater than 44, the low write current level is selected. If the track address is less than 44, the high write current level is selected. Subroutine FINDSE locates the target sector.

The FINDHD and GETDAM subroutines (Figure 5-17) locate the header field and identify the data address mark. The data address mark is a unique combination of data, clocks, and missing clocks for which the microcode searches.

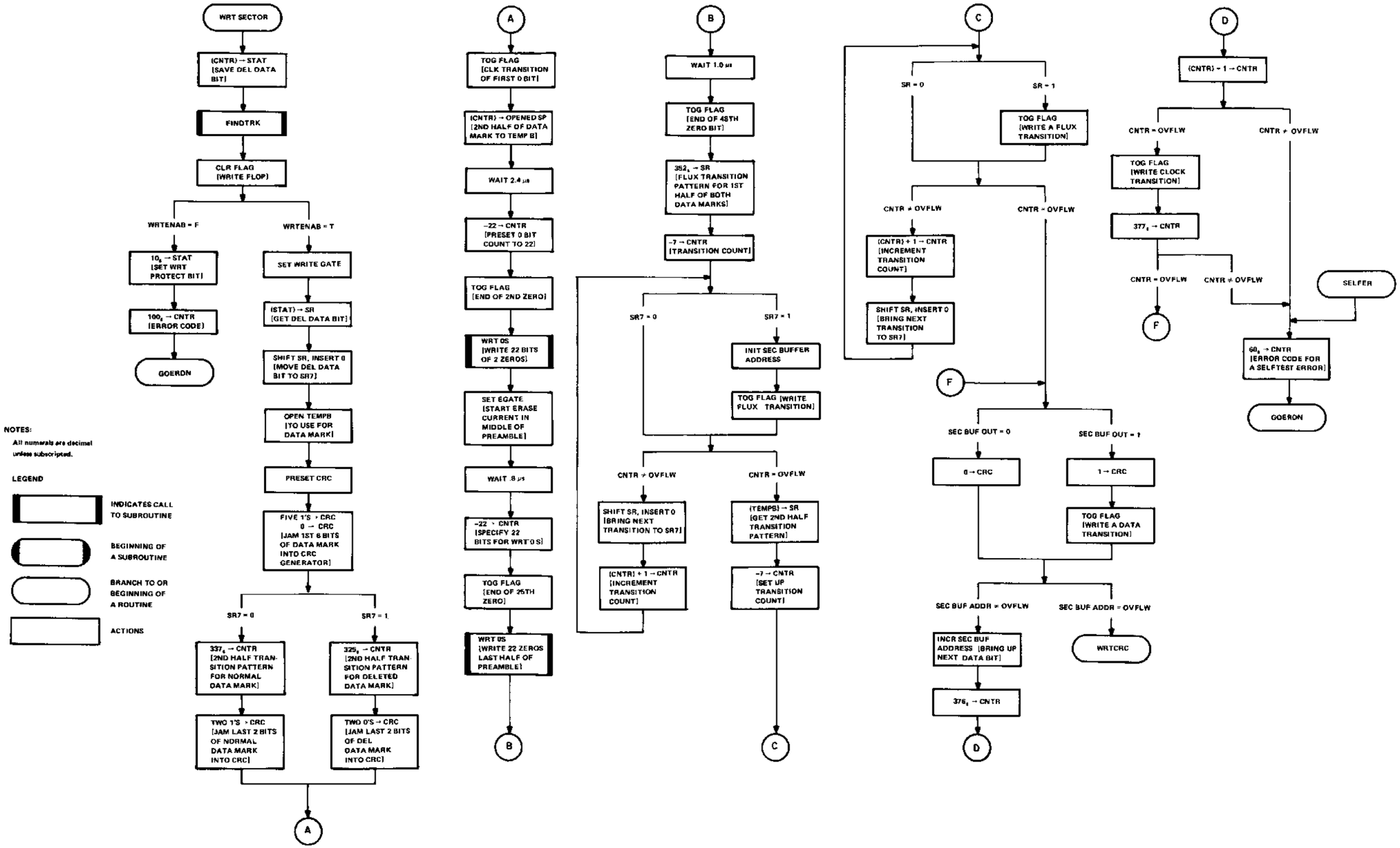
Figure 5-18 contains the routines HDRCOM, BDSRT, and BADHDR, which are continuations of the FINDHD subroutine. The routine HDRCOM is used in comparing a located header with a desired header. A sector address compare and a track compare are made. The BDSRT and BADHDR routines count the number of retries for finding a data mark or an ID address mark. If too many tries are made, the appropriate error codes are set.

Figure 5-19 contains the four subroutines, DELAY, FINDSE, WRT0S, and GETWRD. The DELAY subroutine produces a delay period as set by the delay multiplier that resides in the shift register. The DELAY subroutine is called throughout the microcode to establish waiting times, such as a head load wait of 20 ms. The FINDSE subroutine uses the sector address to locate the correct sector by calling the subroutine FINDHD to find the correct header. The subroutine WRT0S writes the specified number of zeros indicated in the shift register. The GETCMD and GETWRD subroutines differ in that a Transfer Request must be set for a GETWRD. The microcode calls the subroutine WAITRN to wait for a RUN signal from the interface. Error and Done flags are cleared, and either 12 bits or 8 bits are transferred. Parity is checked and the appropriate error code results if a parity error is detected.

Figure 5-20 shows the flowchart for the STEPHD, WAITRN, and MAGCOM subroutine. The STEPHD subroutine moves the head in or out a certain number of tracks as indicated by the counter. When the head is positioned over the desired track, the head is loaded, and a 20 ms delay occurs before the microcode exits from the subroutine. The WAITRN subroutine waits for the RUN signal from the interface. If RUN does not come within 45.87 ms, the head is unloaded, Transfer Request is cleared, and an exit is made out of the subroutine. If RUN does occur within 45.87 ms, Transfer Request is cleared and an exit is made out of the subroutine.

The MAGCOM subroutine compares track addresses and is called by the FINDTR subroutine.

Figure 5-21 shows flowcharts for the DIF and CHKRDY subroutines. The DIF subroutine determines the difference between target track and desired track, so that the STEPHD routine can move to the right track. The CHKRDY subroutine checks for a Drive Ready condition during an Initialize sequence or during a Read Status function.



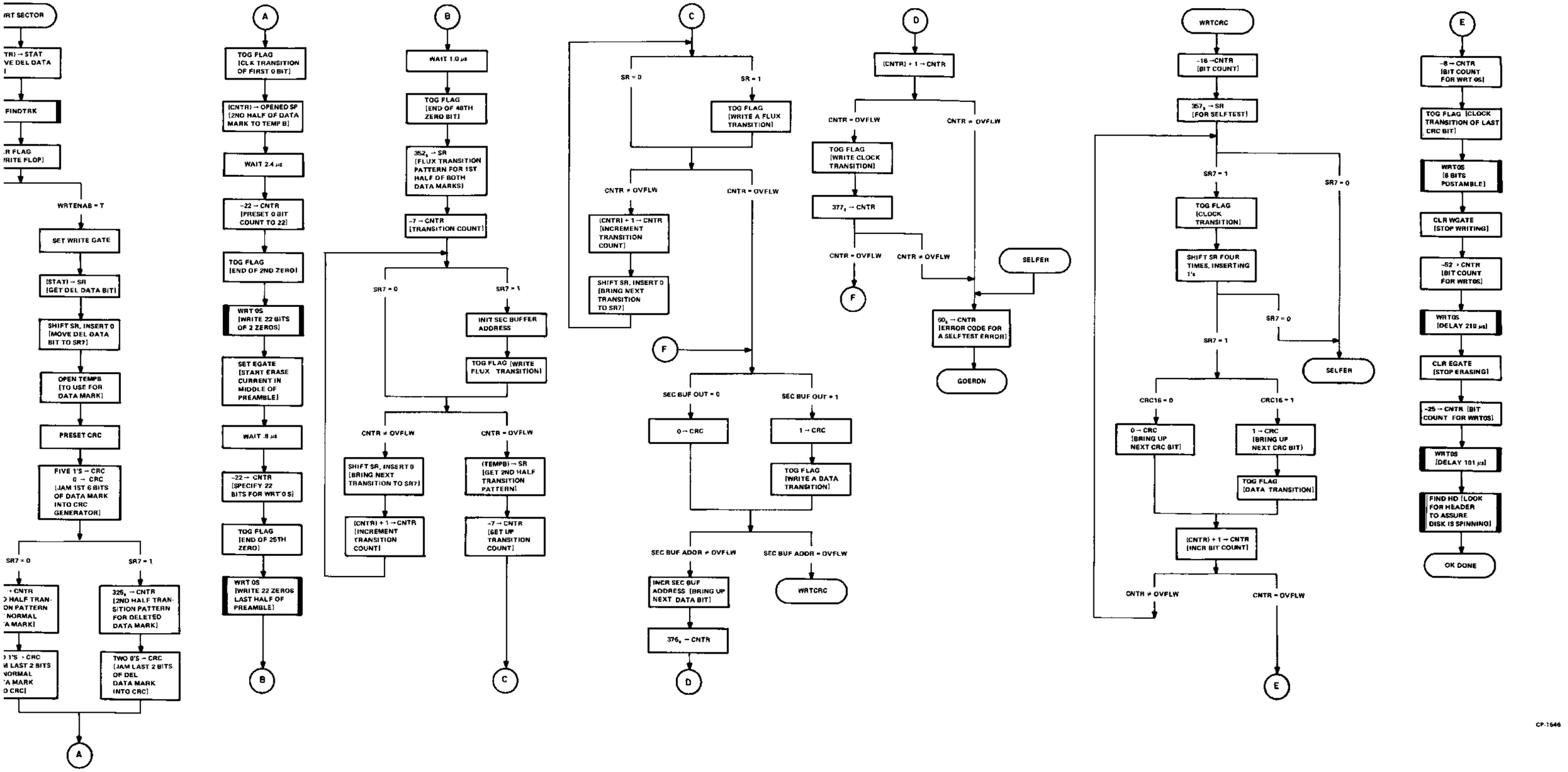
NOTES:

All numerals are decimal unless subscripted.

LEGEND

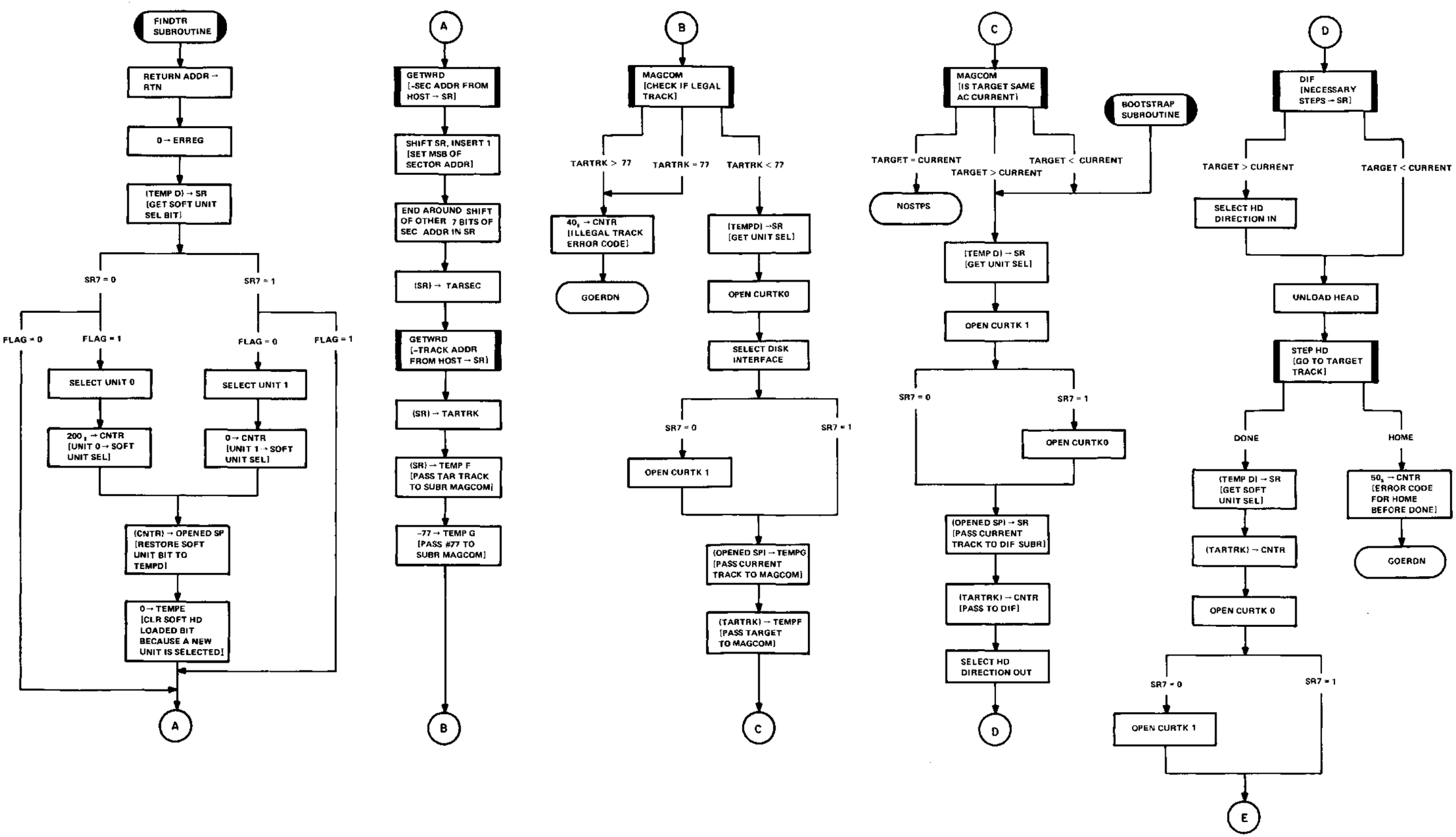
- INDICATES CALL TO SUBROUTINE
- BEGINNING OF A SUBROUTINE
- BRANCH TO OR BEGINNING OF A ROUTINE
- ACTIONS

Figure 5-15 Write Sector Function Flowchart



CP-1646

wchart


FINDTR SUBROUTINE
RETURN ADDR → RTN
0 → ERREG
(TEMP D) → SR (GET SOFT UNIT SEL BIT)
SR7 = 0
SR7 = 1
FLAG = 0
FLAG = 1
FLAG = 0
FLAG = 1
SELECT UNIT 0
SELECT UNIT 1
200 8 → CNTR (UNIT 0 → SOFT UNIT SEL)
0 → CNTR (UNIT 1 → SOFT UNIT SEL)
(CNTR) → OPENED SP (RESTORE SOFT UNIT BIT TO TEMPD)
0 → TEMPE (CLR SOFT HD LOADED BIT BECAUSE A NEW UNIT IS SELECTED)
A
A
GETWRD (-SEC ADDR FROM HOST → SR)
SHIFT SR, INSERT 1 (SET MSB OF SECTOR ADDR)
END AROUND SHIFT OF OTHER 7 BITS OF SEC ADDR IN SR
(SR) → TARSEC
GETWRD (-TRACK ADDR FROM HOST → SR)
(SR) → TARTRK
(SR) → TEMP F (PASS TAR TRACK TO SUBR MAGCOM)
-77 → TEMP G (PASS #77 TO SUBR MAGCOM)
B
B
MAGCOM (CHECK IF LEGAL TRACK)
TARTRK > 77
TARTRK = 77
TARTRK < 77
40 8 → CNTR (ILLEGAL TRACK ERROR CODE)
GOERDN
(TEMPD) → SR (GET UNIT SEL)
OPEN CURTK0
SELECT DISK INTERFACE
SR7 = 0
SR7 = 1
OPEN CURTK 1
(OPENED SP) → TEMPG (PASS CURRENT TRACK TO MAGCOM)
(TARTRK) → TEMPF (PASS TARGET TO MAGCOM)
C
C
MAGCOM (IS TARGET SAME AC CURRENT)
BOOTSTRAP SUBROUTINE
TARGET = CURRENT
TARGET > CURRENT
TARGET < CURRENT
NOSTPS
(TEMP D) → SR (GET UNIT SEL)
OPEN CURTK 1
SR7 = 0
SR7 = 1
OPEN CURTK0
(OPENED SP) → SR (PASS CURRENT TRACK TO DIF SUBR)
(TARTRK) → CNTR (PASS TO DIF)
SELECT HD DIRECTION OUT
D
D
DIF (NECESSARY STEPS → SR)
TARGET > CURRENT
TARGET < CURRENT
SELECT HD DIRECTION IN
UNLOAD HEAD
STEP HD (GO TO TARGET TRACK)
DONE
HOME
(TEMP D) → SR (GET SOFT UNIT SEL)
(TARTRK) → CNTR
OPEN CURTK 0
50 8 → CNTR (ERROR CODE FOR HOME BEFORE DONE)
GOERDN
SR7 = 0
SR7 = 1
OPEN CURTK 1
E
E

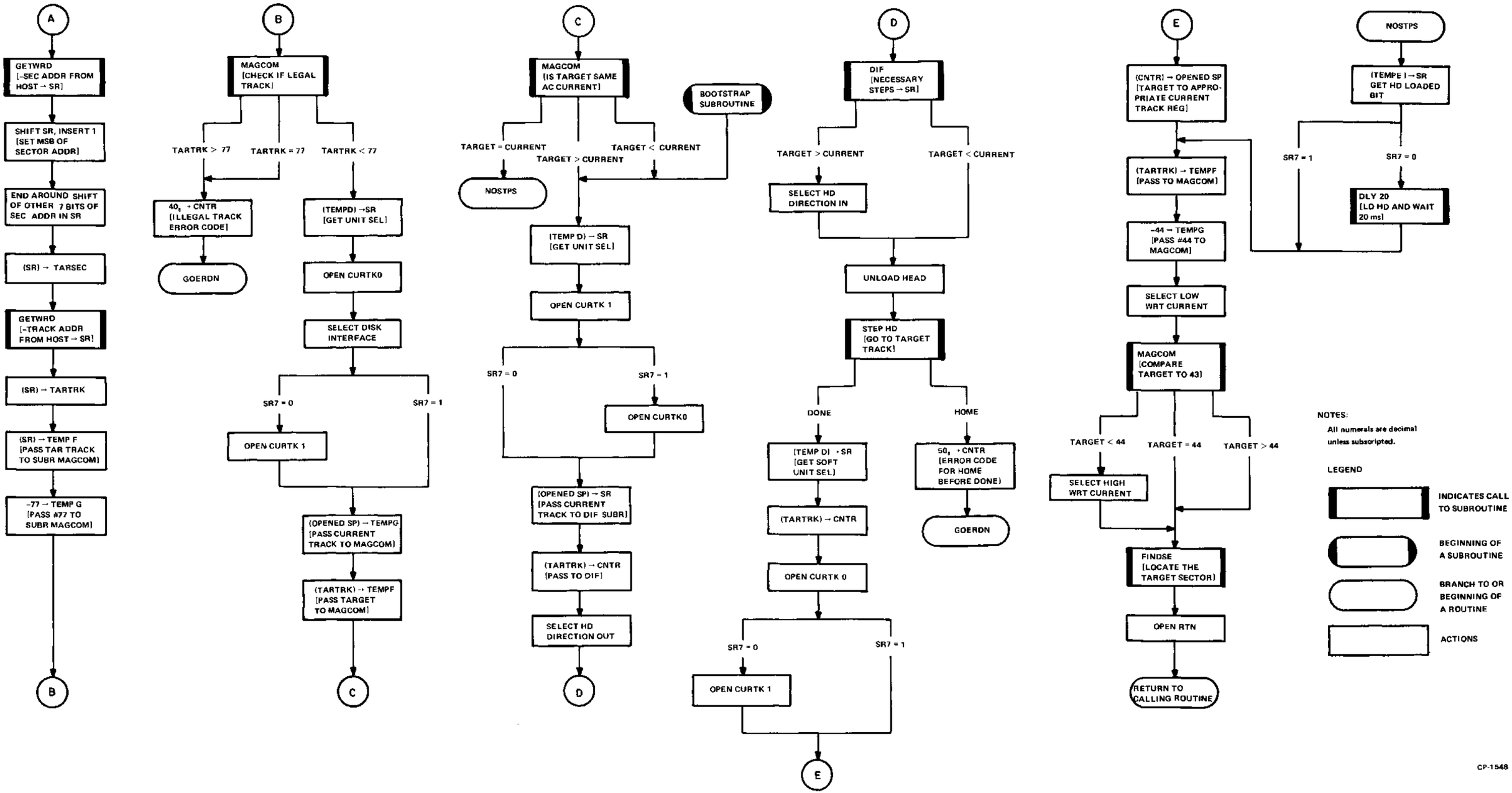


Figure 5-16 FINDTR Subroutine Flowchart

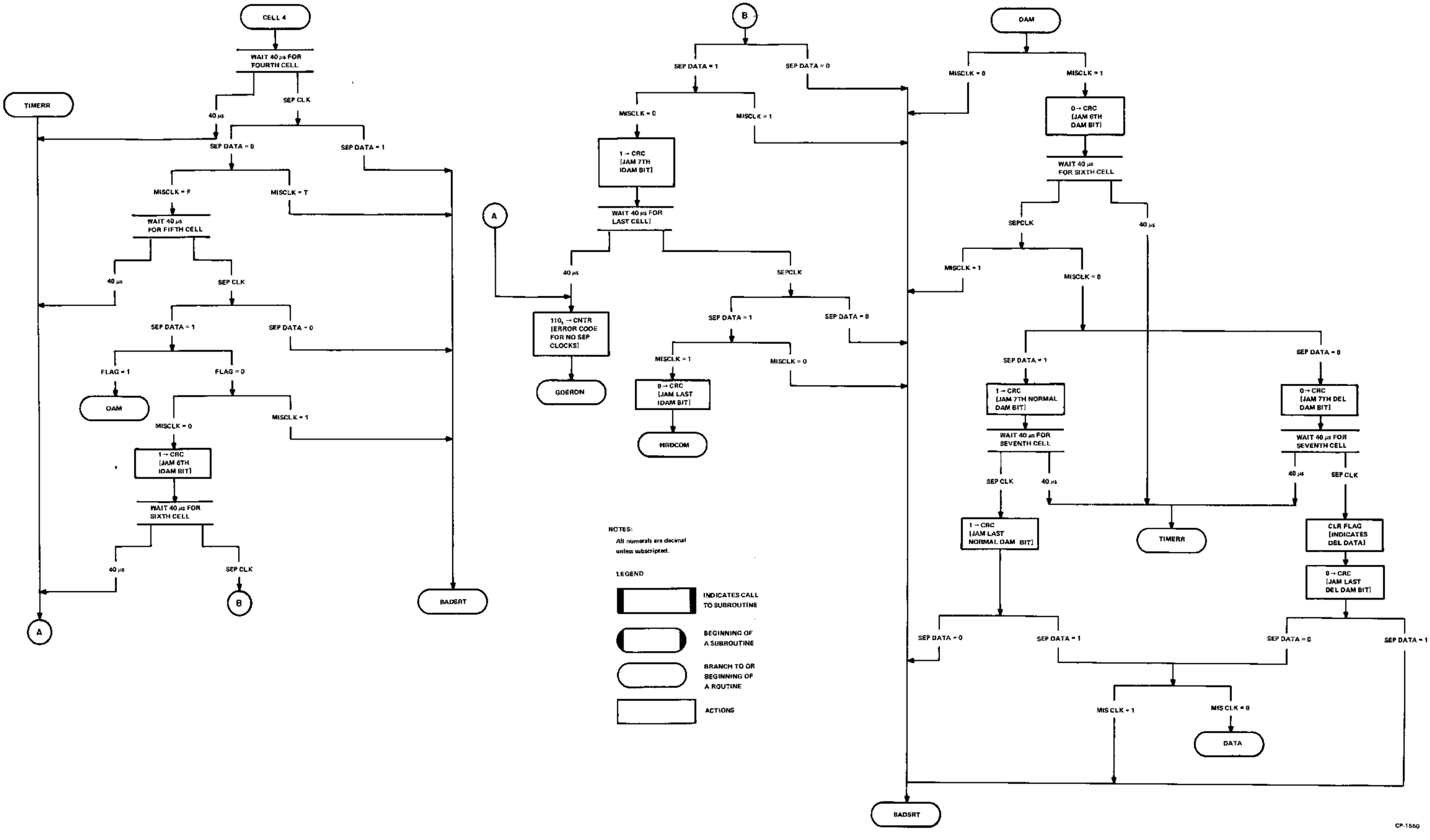
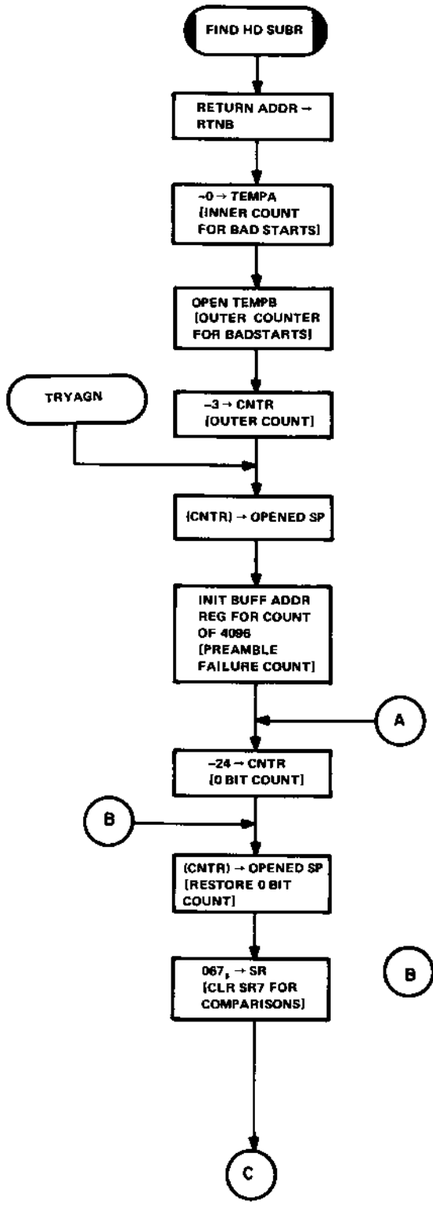
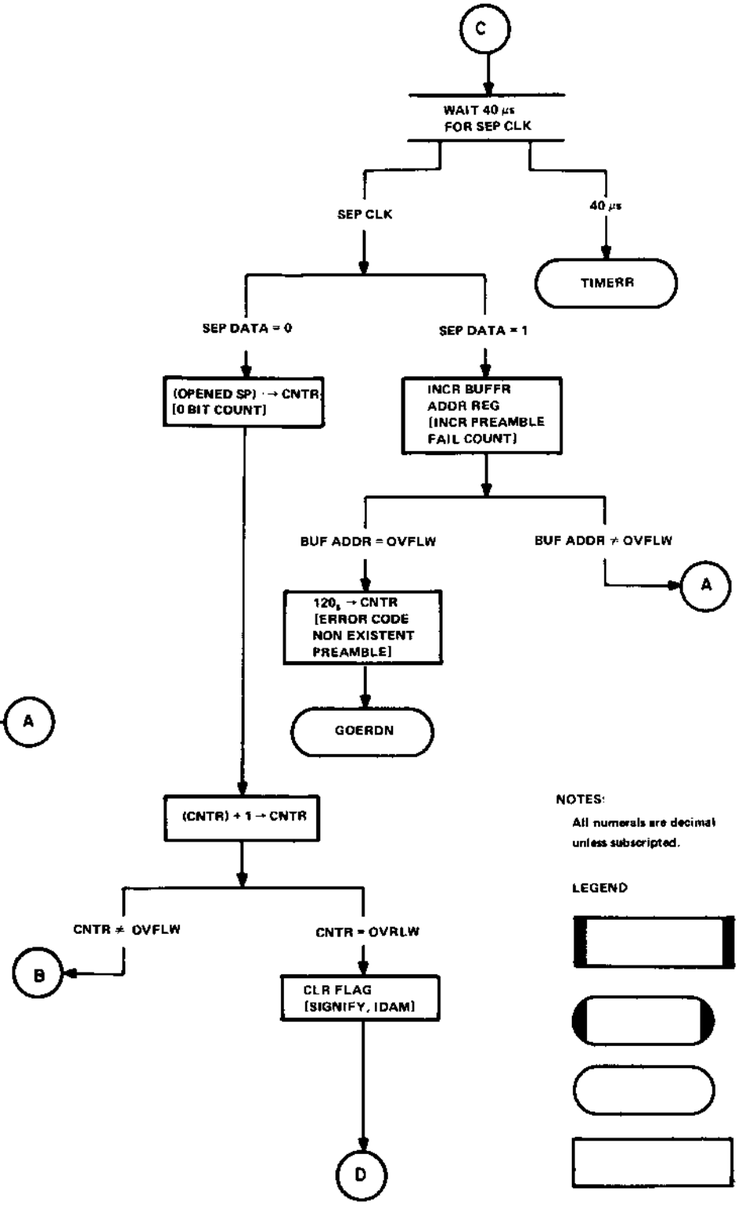
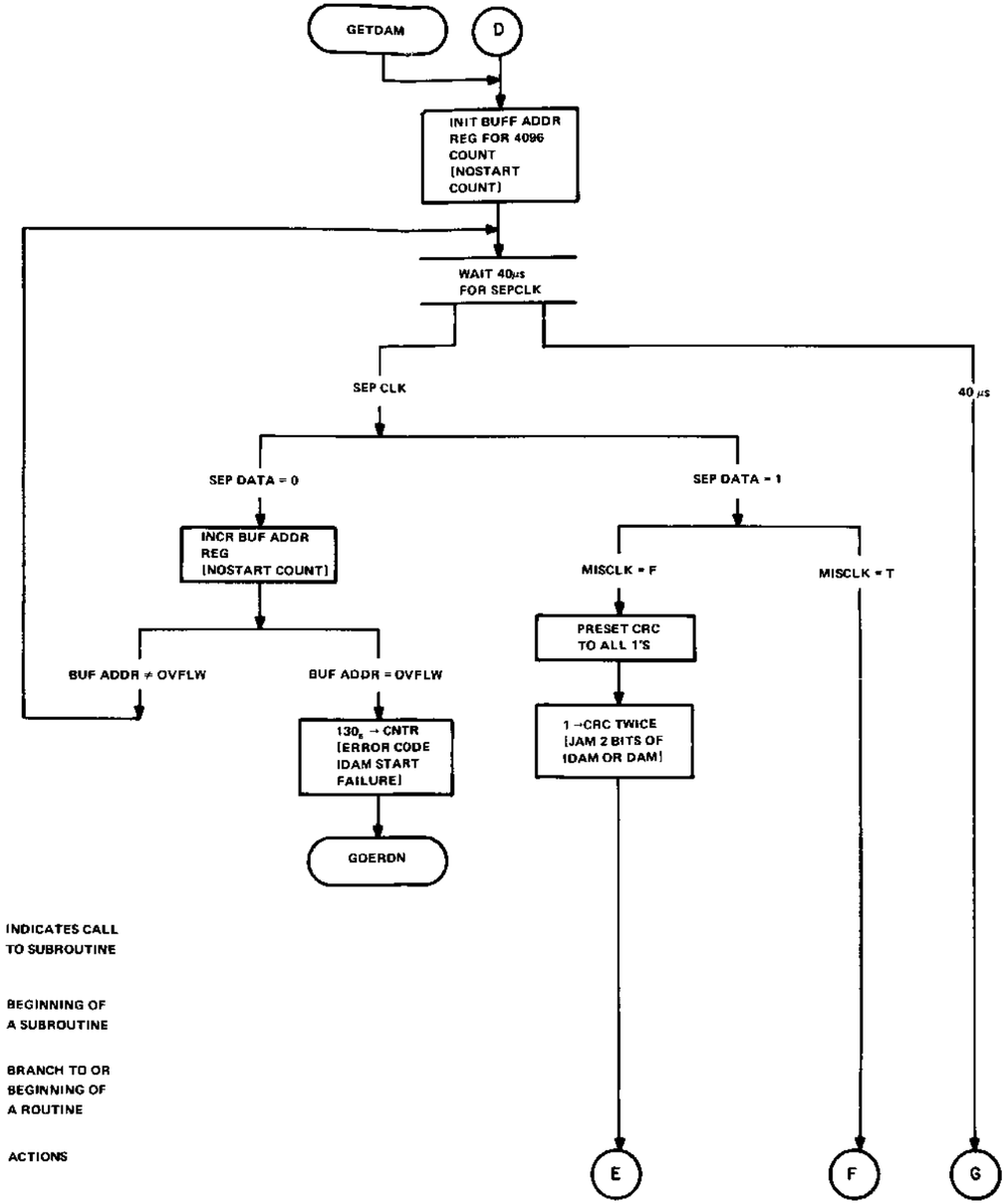
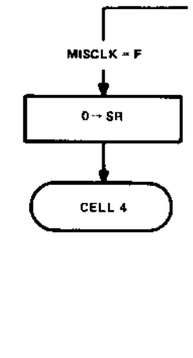


Figure 5-17 FINDHD and GETDAM Subroutines Flowchart (Sheet 1 of 2)









NOTES:

All numerals are decimal unless subscripted.

LEGEND

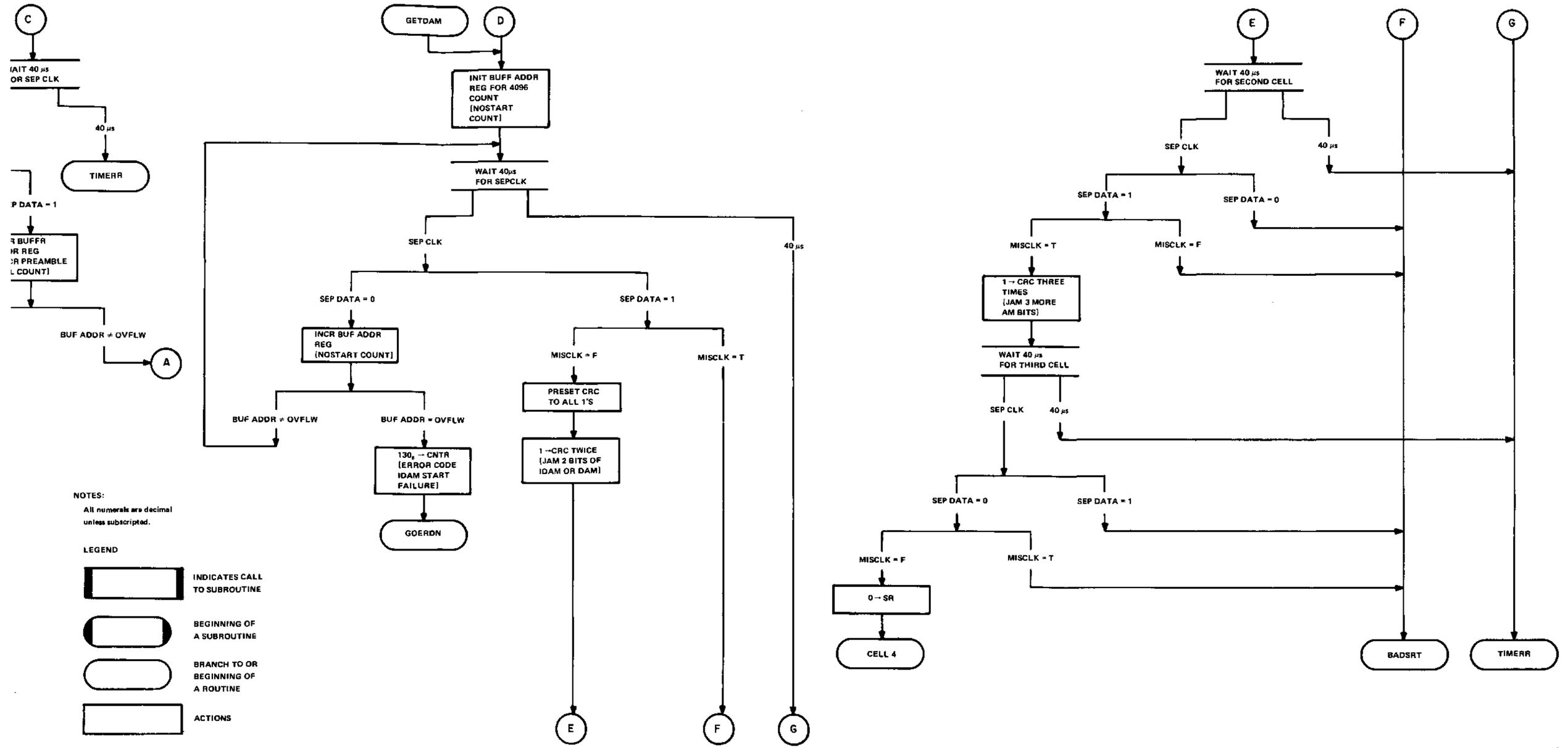
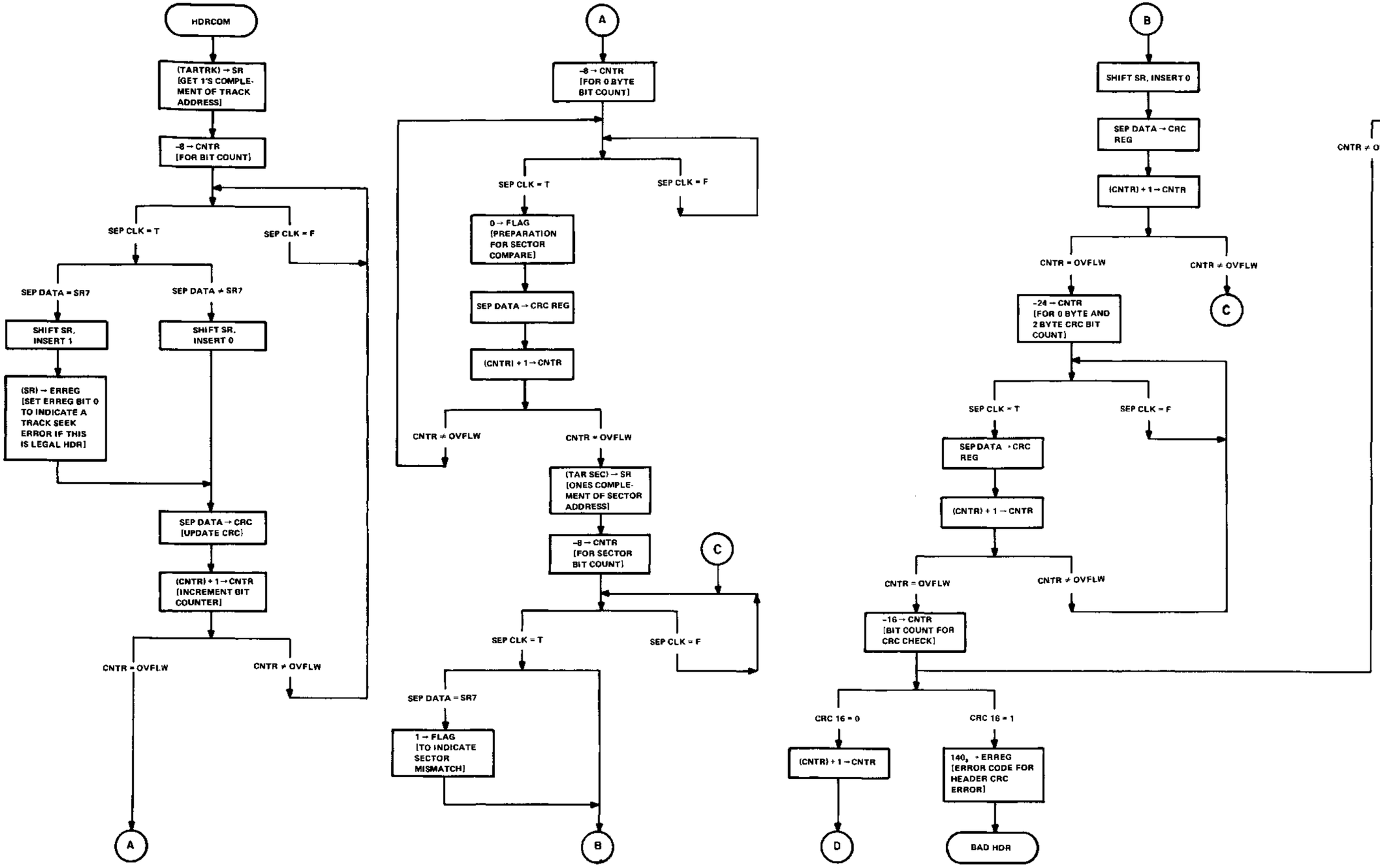


Figure 5-17 FINDHD and GETDAM Subroutines Flowchart (Sheet 2 of 2)



NOTES:
All numerals
unless subsc

LEGEND

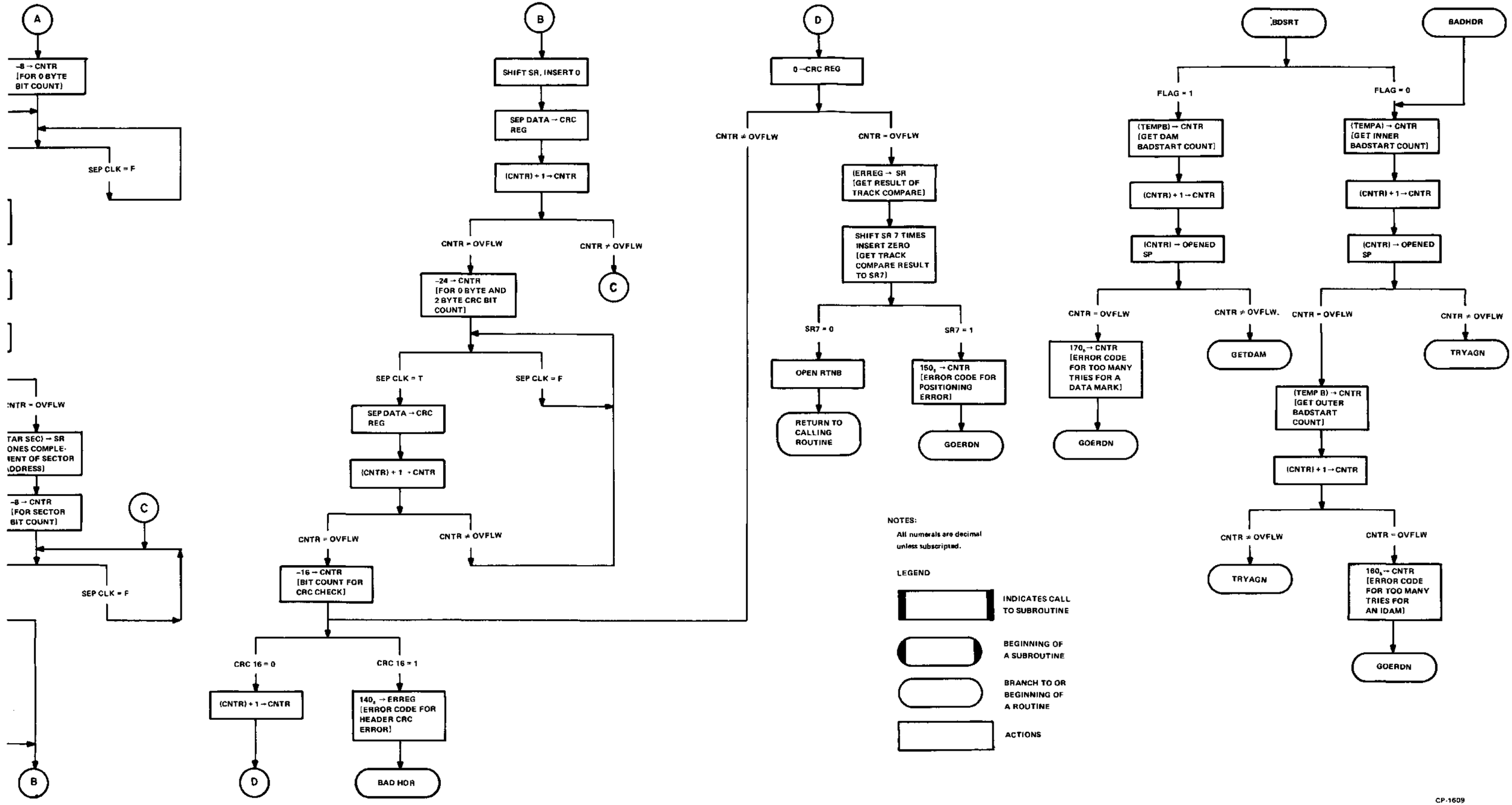


Figure 5-18 HDRCOM, BDSRT, BADHDR Routines Flowchart

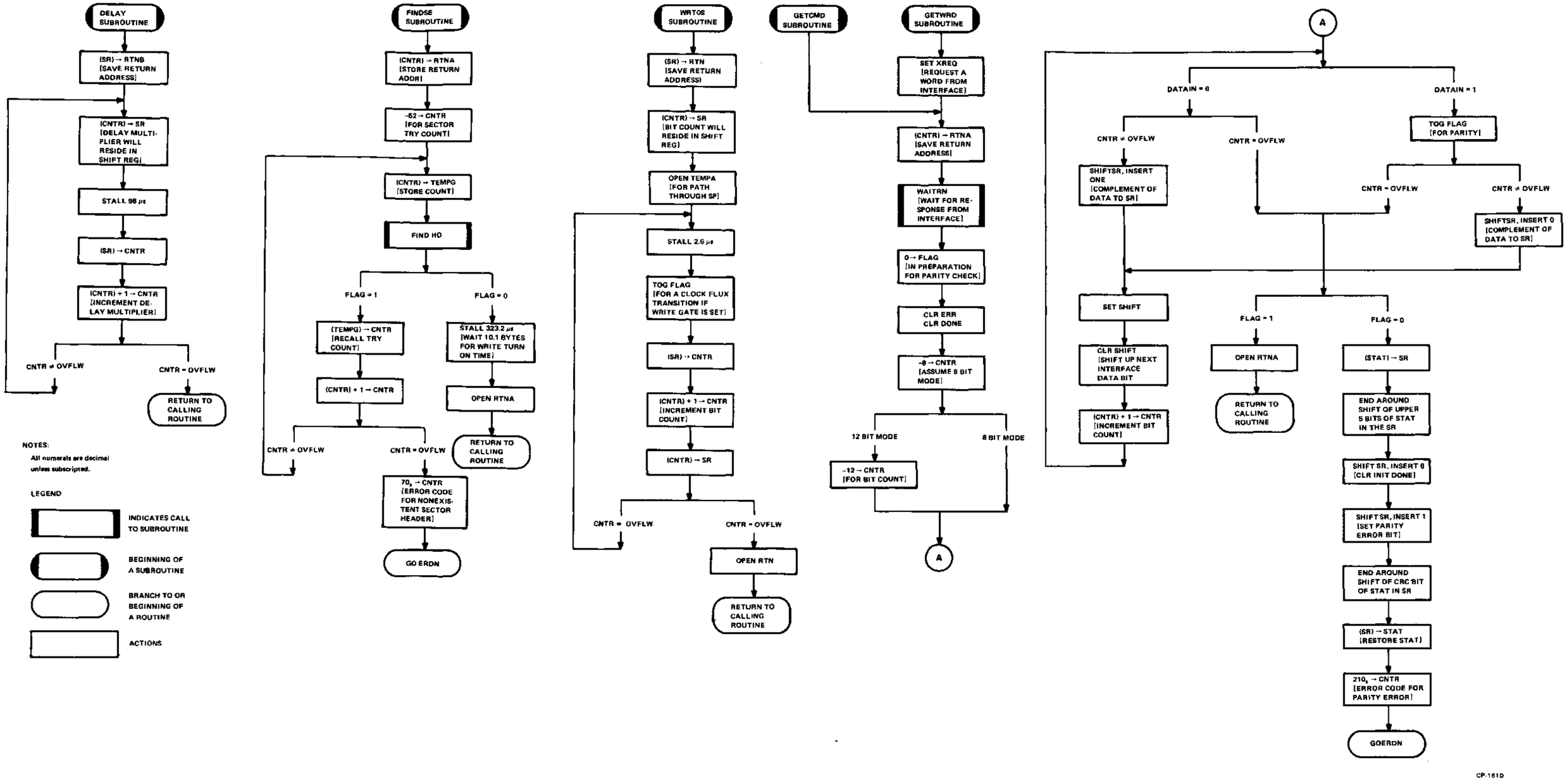


Figure 5-19 DELAY, FINDSE, WRTOS, GETWRD Subroutines Flowchart

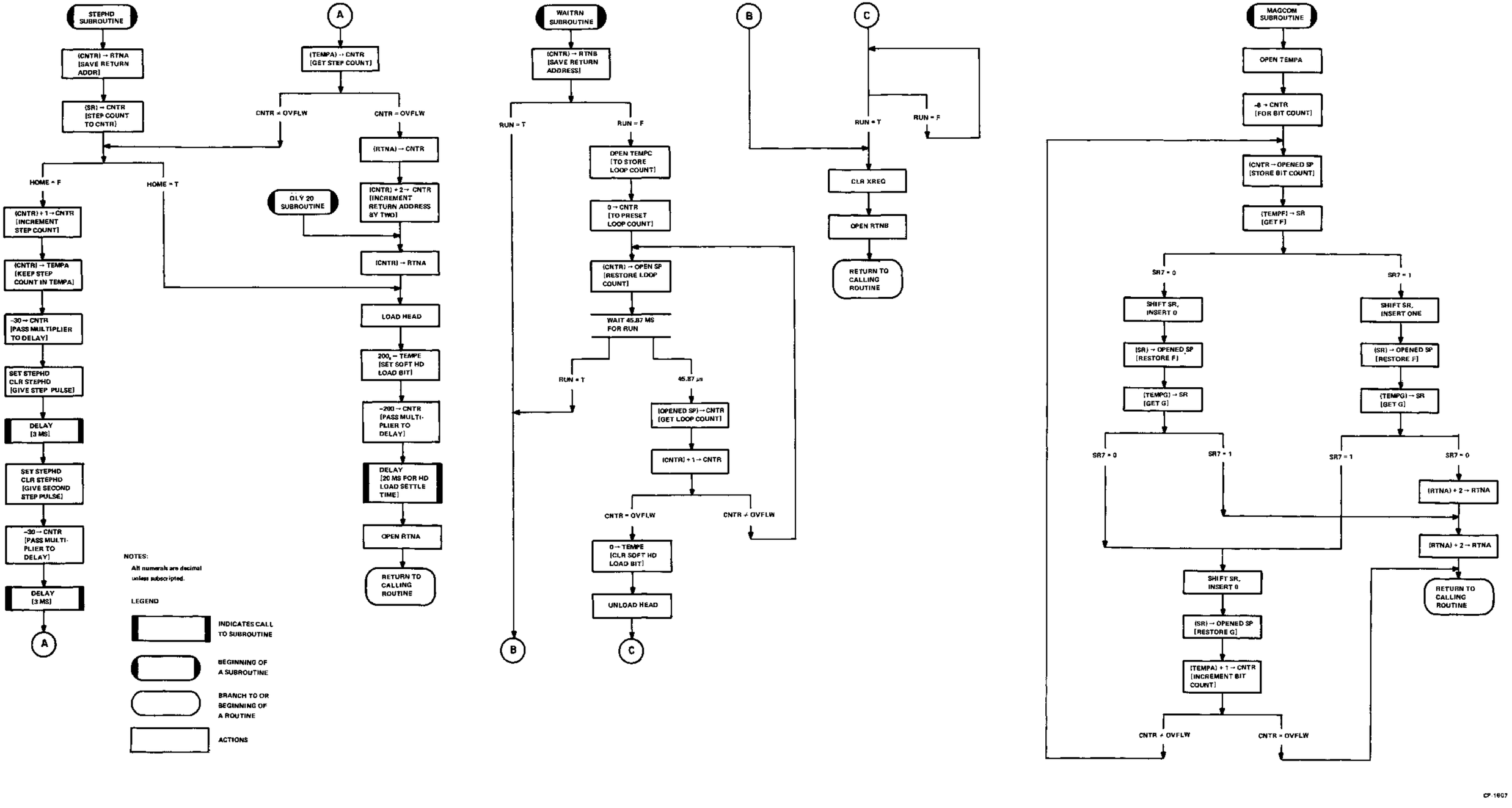


Figure 5-20 STEPHD, WAITRN, MAGCOM Subroutines Flowchart

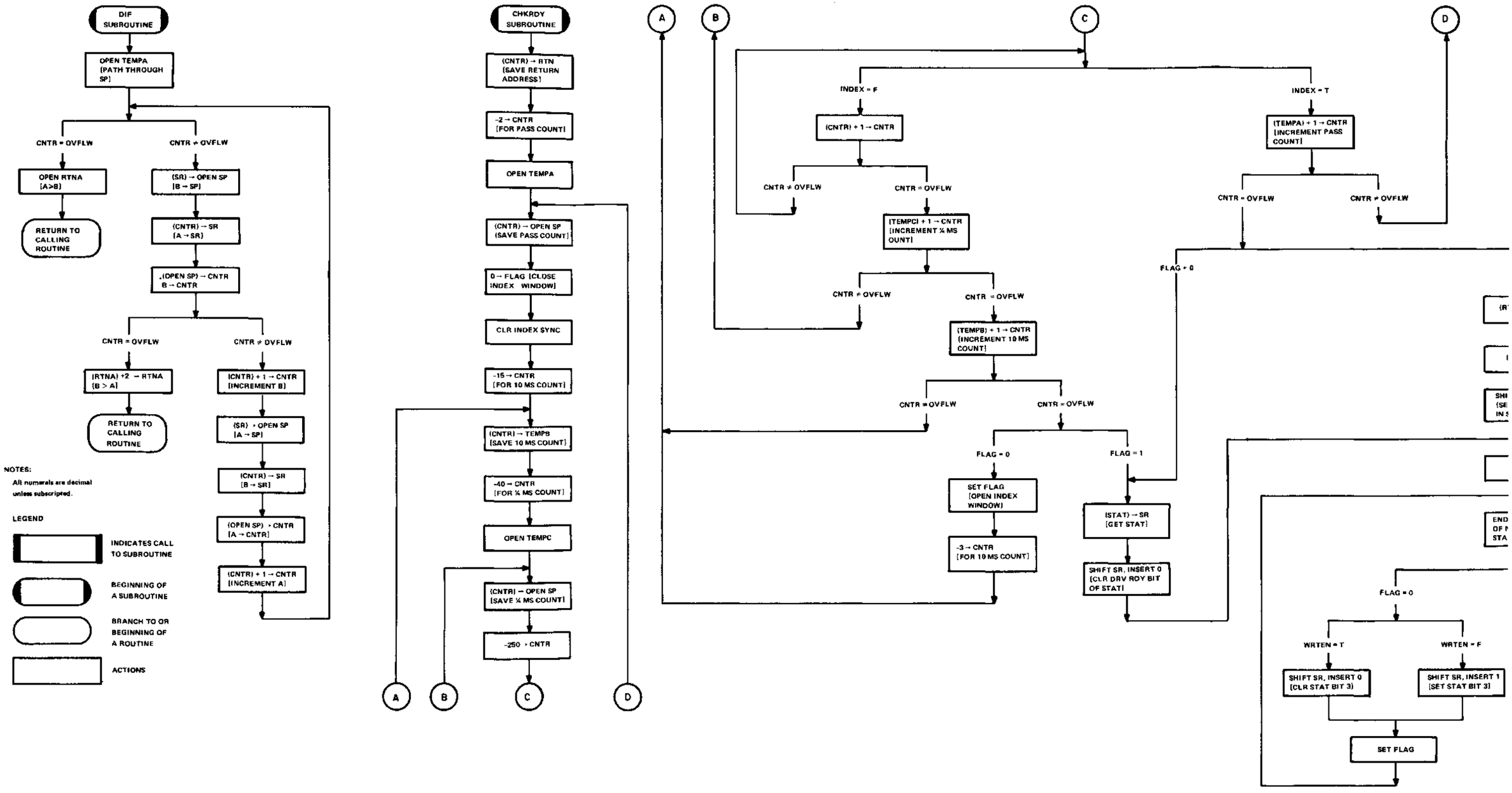
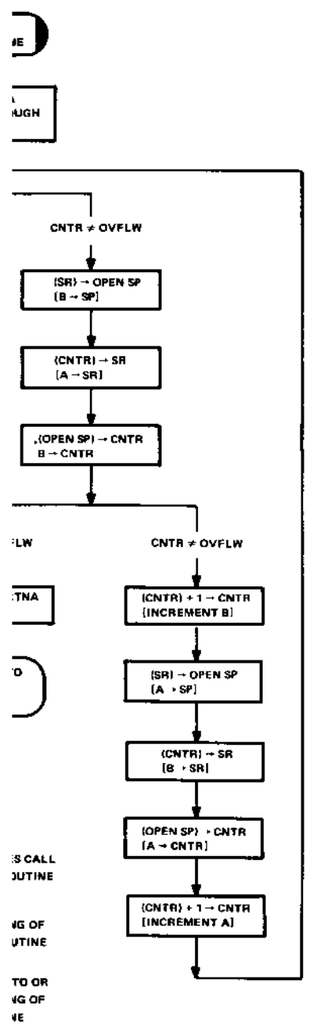
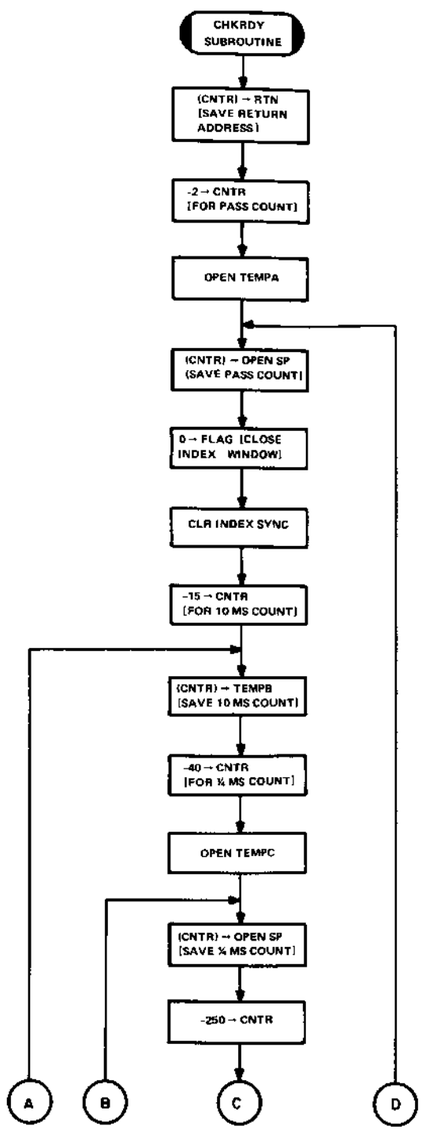
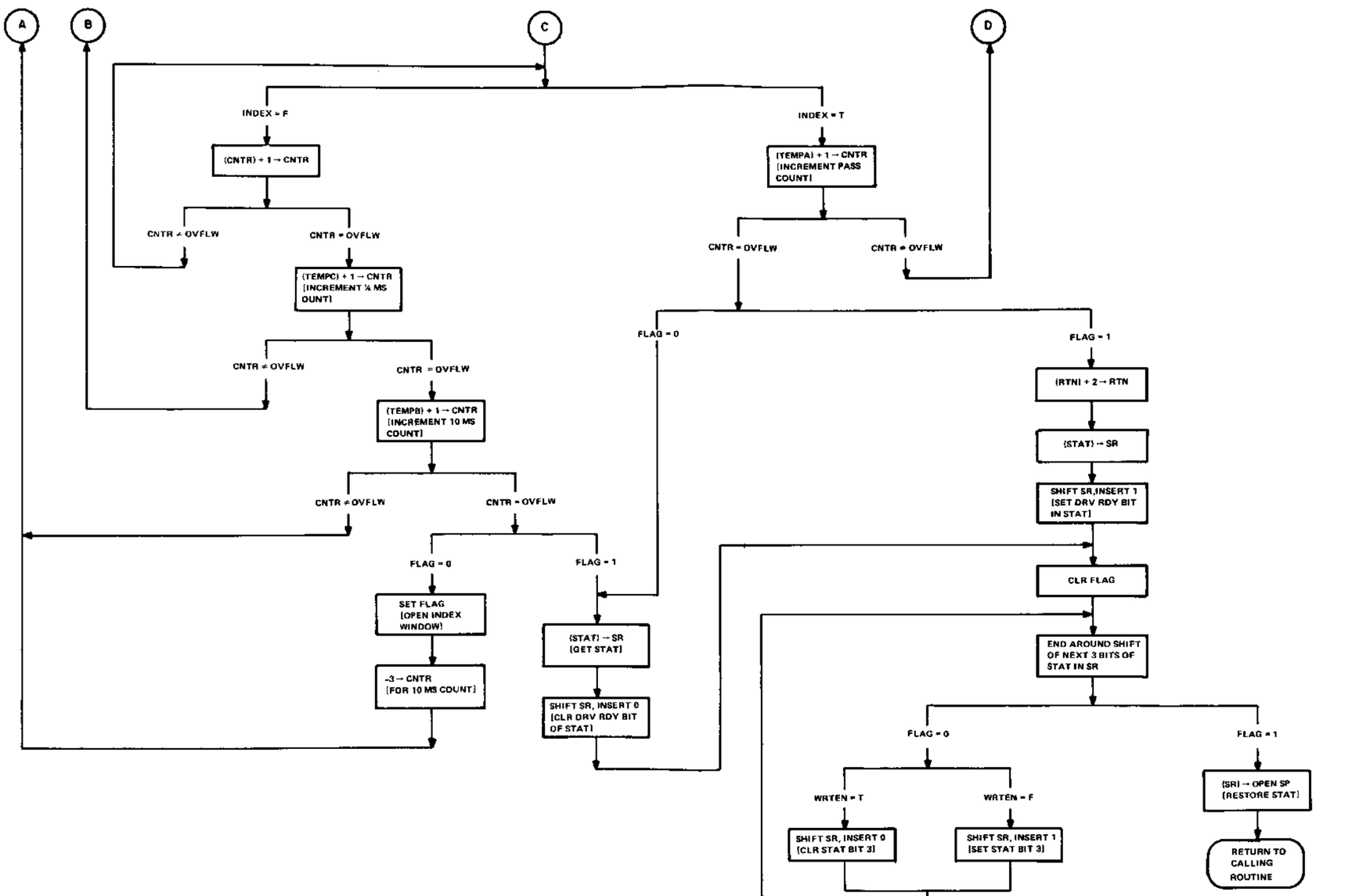

DIF SUBROUTINE
OPEN TEMPA (PATH THROUGH SP)
CNTR = OVFLW
CNTR ≠ OVFLW
OPEN RTNA (A>B)
RETURN TO CALLING ROUTINE
(SR) → OPEN SP (B → SP)
(CNTR) → SR (A → SR)
(OPEN SP) → CNTR B → CNTR
CNTR = OVFLW
CNTR ≠ OVFLW
(RTNA) +2 → RTNA (B > A)
RETURN TO CALLING ROUTINE
(CNTR) + 1 → CNTR (INCREMENT B)
(SR) → OPEN SP (A → SP)
(CNTR) → SR (B → SR)
(OPEN SP) → CNTR (A → CNTR)
(CNTR) + 1 → CNTR (INCREMENT A)
NOTES:
All numerals are decimal unless subscripted.
LEGEND
INDICATES CALL TO SUBROUTINE
BEGINNING OF A SUBROUTINE
BRANCH TO OR BEGINNING OF A ROUTINE
ACTIONS
CHKRDY SUBROUTINE
(CNTR) → RTN (SAVE RETURN ADDRESS)
-2 → CNTR (FOR PASS COUNT)
OPEN TEMPA
(CNTR) → OPEN SP (SAVE PASS COUNT)
0 → FLAG (CLOSE INDEX WINDOW)
CLR INDEX SYNC
-15 → CNTR (FOR 10 MS COUNT)
(CNTR) → TEMPB (SAVE 10 MS COUNT)
-40 → CNTR (FOR ¼ MS COUNT)
OPEN TEMPC
(CNTR) → OPEN SP (SAVE ¼ MS COUNT)
-250 → CNTR
A
B
C
D
INDEX = F
(CNTR) + 1 → CNTR
CNTR ≠ OVFLW
CNTR = OVFLW
(TEMPC) + 1 → CNTR (INCREMENT ¼ MS OUNT)
CNTR ≠ OVFLW
CNTR = OVFLW
(TEMPB) + 1 → CNTR (INCREMENT 10 MS COUNT)
CNTR ≠ OVFLW
CNTR = OVFLW
FLAG = 0
FLAG = 1
SET FLAG (OPEN INDEX WINDOW)
-3 → CNTR (FOR 10 MS COUNT)
(STAT) → SR (GET STAT)
SHIFT SR, INSERT 0 (CLR DRV RDY BIT OF STAT)
INDEX = T
(TEMPA) + 1 → CNTR (INCREMENT PASS COUNT)
CNTR = OVFLW
CNTR ≠ OVFLW
FLAG = 0
FLAG = 0
WRTEN = T
WRTEN = F
SHIFT SR, INSERT 0 (CLR STAT BIT 3)
SHIFT SR, INSERT 1 (SET STAT BIT 3)
SET FLAG

Figure 5-21







CP-1508

Figure 5-21 DIF and CHKRDY Subroutine Flowchart

5.2.6 Read/Write Electronics

A detailed block diagram of the read/write electronics is shown in Figure 5-22. The read/write electronics can be separated into four functions:

1. Diskette position
2. Head read/write circuitry
3. Head load control and solenoid drivers
4. Stepper motor control and motor drivers

5.2.6.1 Diskette Position – Track 0 and index hole detection are accomplished by an LED-phototransistor pair. There are separate circuits for drive 0 and drive 1 for a total of four sensing circuits. A schematic diagram of the circuitry appears on page D4 of the RX8/RX11 Print Sets.

The six sensor indicator lines are input to a 74157 data selector shown on page D6, which outputs track 0 and index hole sensing information to the μCPU controller. The 74157 is data-selected by the SEL 1 H signal from the μCPU controller, indicating which drive is being used.

5.2.6.2 Head Read/Write Circuitry – The head read/write circuitry is shown on page D3 and is composed of a write section and a read section.

Head writing is controlled by six signals from the μCPU controller as shown in Figure 5-22. Each of these signals is further explained in Paragraph 5.1.4. During a write operation, D6 WT GATE H initiates a Write command, and D6 ERASE H activates the tunnel erase drivers. The data stream D6 WT DATA H is amplified by the head write current amplifiers and directed to the proper drive by D6 SEL DK0 H and D6 SEL DK1 H signals. Head writing is inhibited by D6 SEL WT PROT L or D6 DC LO L signals. The signal ABOVE TK 43 H controls proper write current to the heads.

Head reading is accomplished by the circuitry shown on page D3. The diode and resistor circuitry on the R/W BUS + and R/W BUS - protect the 733 preamplifier during a write operation. The filter following the preamplifier eliminates unwanted head noise. The next 733 amplifier stage drives the peak detection circuitry, composed of a differentiator and crossover detector (1414). The output of the 1414 is fed to 74123s (monostables), which are used as pulse shapers. RAW DATA L, which represents digitized data from the diskette, is sent to the μCPU controller.

5.2.6.3 Head Load Control and Solenoid Drivers – Head load control and solenoid driver circuitry is shown on page D6. A D6 LOAD HEAD H signal from the μCPU controller causes either head load solenoid to be activated, depending on whether signal D6 SEL DK1 H is high or low. D5 INIT L, an initialize signal from the stepper motor control, will reset the drive and cause the head to unload.

5.2.6.4 Stepper Motor Control and Motor Drivers – Stepper motor control and motor driver circuitry is shown on page D5. A separate and identical control section and motor driver section exists for each drive. Signals D6 SEL DK1 H and D6 SEL DK0 H determine which drivers to activate.

D6 OUT H determines the direction of movement of the head in response to the pulse D6 STEP L from the μCPU controller. If D6 OUT H is asserted, the head moves outward toward track 0; if it is negated the head moves inward toward track 77. The two 7473s and the 7450 in the two control sections are connected as up/down counters to control the four motor driver transistors. In moving OUT, the counter counts up to turn on the driver transistors to move the head outward. In moving IN, the counter counts down to turn on the driver transistors to move the head inward. Each track position requires two phases of the counter. Therefore, two step pulses are required for each track moved.

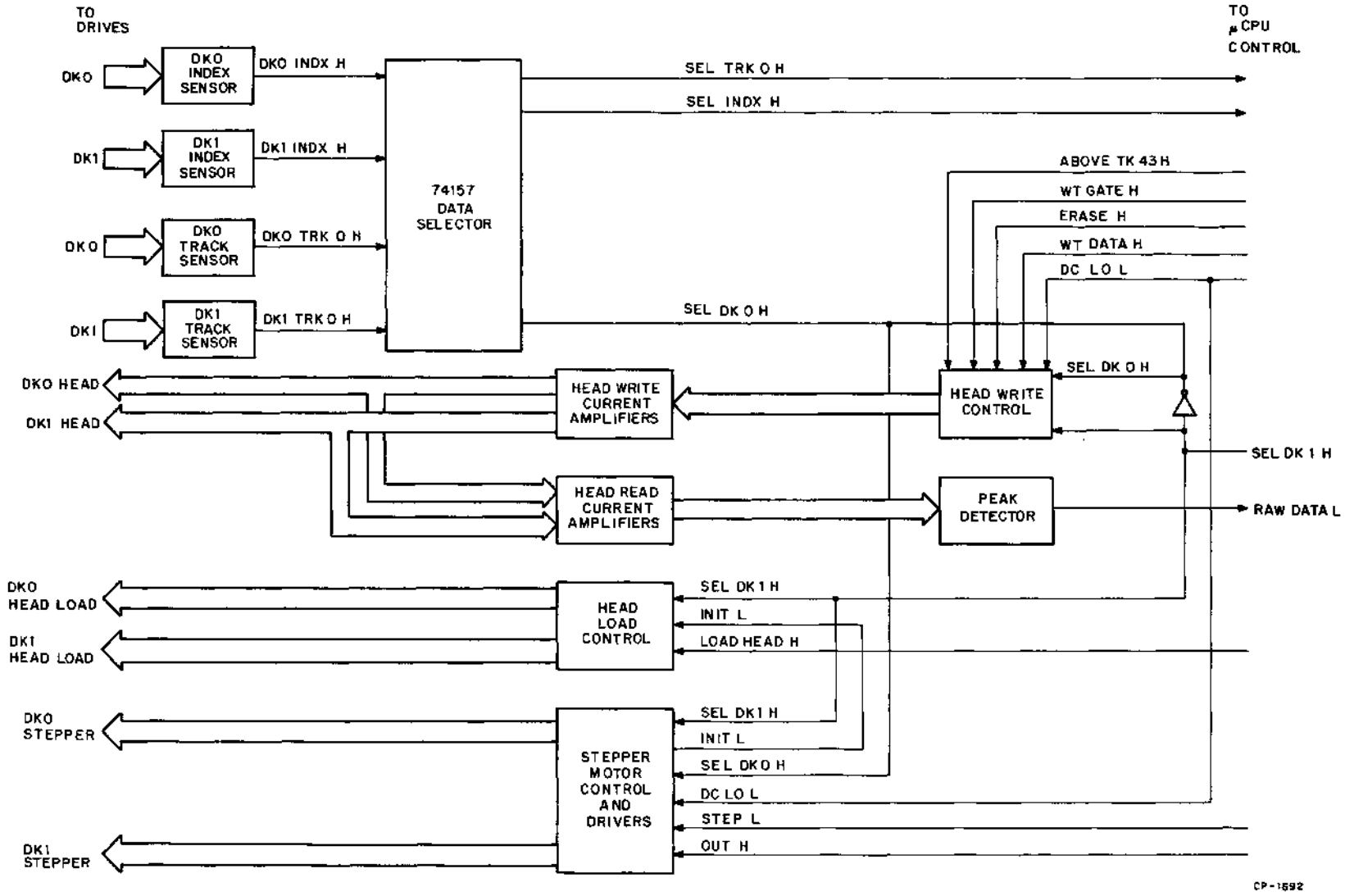


Figure 5-22 Read/Write Electronics Block Diagram

5.2.7 Mechanical Drive

The mechanical drive consists of four major parts:

1. Drive mechanism
2. Spindle mechanism
3. Positioning mechanism
4. Head load mechanism

The complete mechanical structure of the drive is shown in Figure 5-23, and each section is described in the following paragraphs.

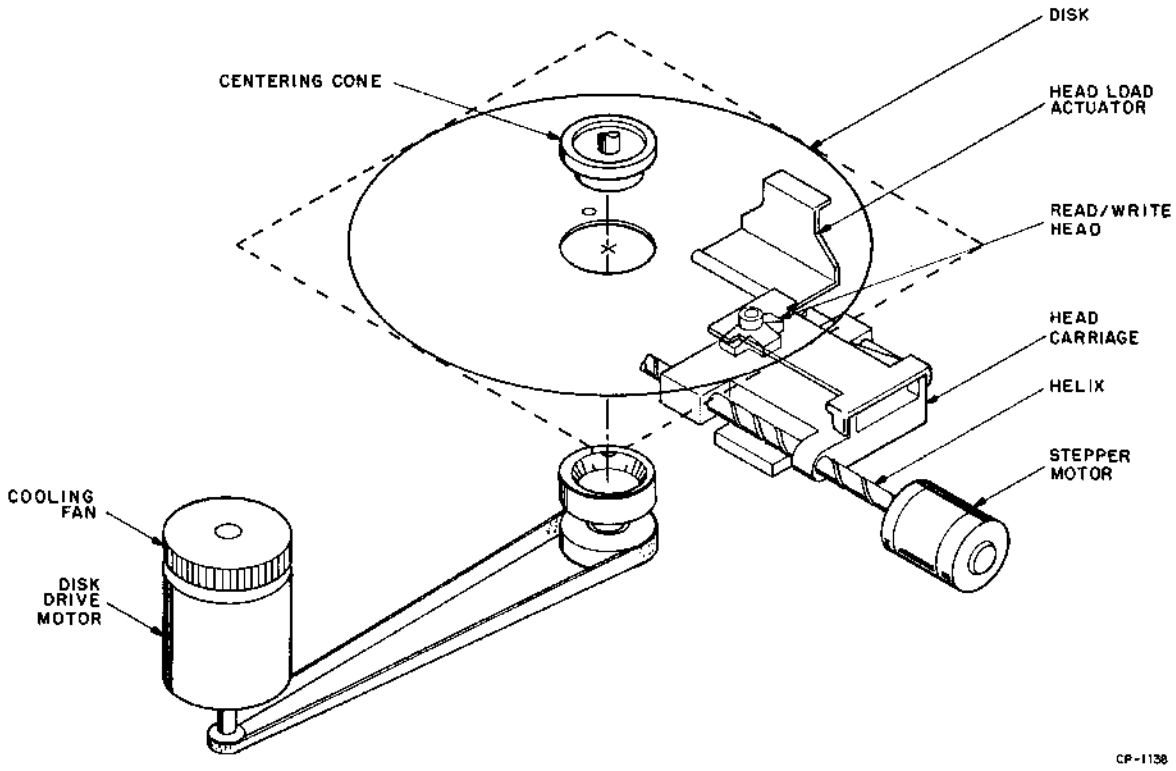


Figure 5-23 Disk Drive Mechanical System

5.2.7.1 Drive Mechanism – The drive system provides rotational diskette movement using a single-phase motor selected to match primary power of the controller system (Figure 5-24). The diskette drive attains ready status within 2 seconds of primary power application.

A cooling fan is mounted on one end of the drive motor shaft. Rotation of the diskette is provided by a belt and pulley connected to the other end of the motor shaft. The drive pulley and belt are selected for either 50 or 60 Hz power to achieve a diskette rotational speed of 360 rpm. See Paragraph 2.2.3.2 for complete input power modification requirements.

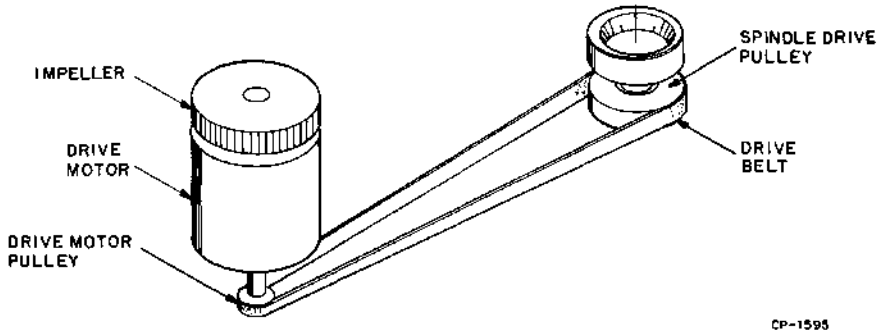


Figure 5-24 Drive Mechanism

5.2.7.2 Spindle Mechanism – The spindle mechanism consists of a centering cone and a load plate. In the unload position, the load plate is pivoted upward, creating an aperture through which the floppy diskette is inserted. In this position, the centering cone disengages the diskette from the drive mechanism.

To load a diskette, the operator inserts the floppy diskette and presses down on the load handle, which latches the load plate in the operating mode. The centering cone is mechanically linked to the load plate and is activated at the same time. (Figure 5-25).

The centering cone is an open splined nylon device that performs two functions:

1. Engages the diskette media and drive mechanism.

2. Positions the diskette media in the correct track alignment.

As the load plate is pivoted to the load position, the centering cone enters the floppy diskette center. At approximately 80 mils from the fully down position, a centering cone expander is automatically activated. This device then expands the centering cone, which grips the inner diameter of the diskette media in the correct track alignment.

The track 0 position serves as the diskette drive reference track. This position is sensed by a phototransducer, which generates track 0 status. This status is sent to the controller for initial track positioning. The controller generates step pulses to position the carriage from the current track to a new track.

5.2.7.3 Positioning Mechanism – The positioning mechanism comprises a carriage assembly and a bidirectional stepper motor (Figure 5-26). The stepper motor rotational movements are converted to linear motion by the rotor helix drive.

The read/write head mount rides in the grooved helix shaft and is held in horizontal alignment by the way. When the stepper motor is pulsed, the helix drive rotates clockwise or counterclockwise, moving the mount in or out.

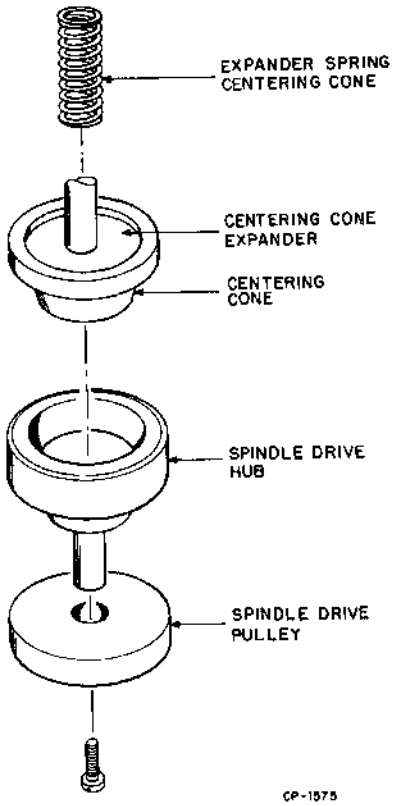


Figure 5-25 Centering Cone and Drive Hub

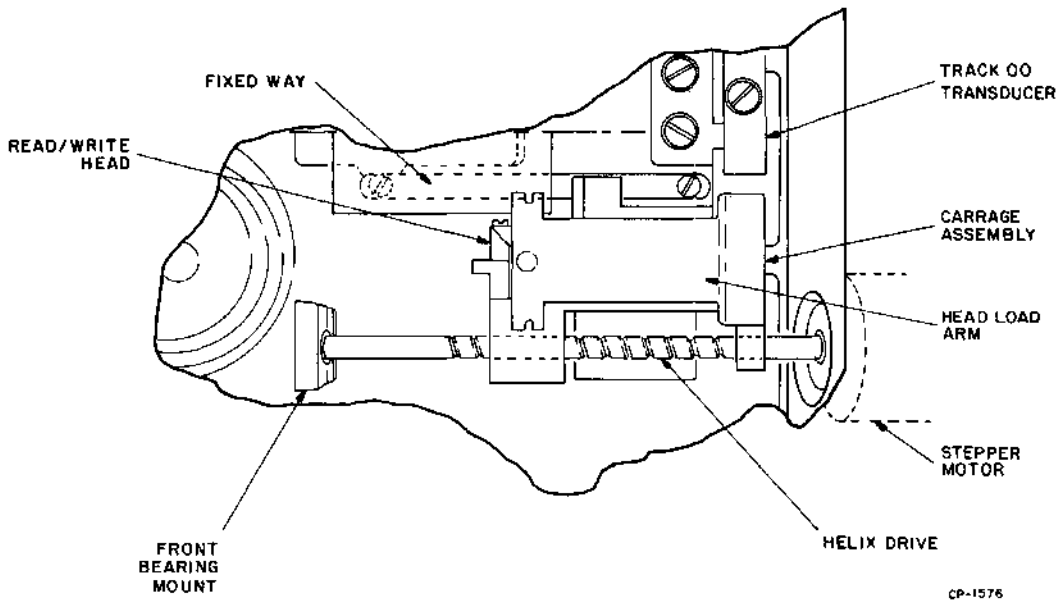


Figure 5-26 Positioning Mechanism

The stepper motor includes four pairs of quadrature windings. In detent, current flows in one winding and maintains the rotor in electro-magnetic detent. For positioning, one or more step pulses are sequentially applied to quadrature windings, causing an imbalance in the electro-magnetic field. Consequently, the stepper motor rotor revolves through detent positions until the step pulses are halted. The rotor then locks in that position. The sequence in which the stepper motor quadrature windings are pulsed dictates rotational direction and, subsequently, higher or lower track addressing from a relative position.

5.2.7.4 **Head Load Mechanism** – The head load mechanism is basically a relay driver and a solenoid. When activated by signal LD HD from the controller, the spring-loaded head load pad is released and rests in parallel alignment with the floppy diskette surface. Part of the casting provides the lower alignment dimensional surface, while the head load solenoid bar provides the upper alignment surface.

In the load position, the read/write head tang rides between these two alignment surfaces and keeps the read/write head in contact with the diskette surface. The load pad is located behind the read/write head and holds the floppy diskette flat against the lower alignment block.

To minimize diskette surface and head wear, the head is automatically disabled by the controller if no new command has been issued within 48 ms. Head settling time is 20 ms.

CHAPTER 6
MAINTENANCE

6.1 RECOMMENDED TOOLS AND TEST EQUIPMENT

Table 6-1 lists the recommended tools and test equipment for maintenance of the RX8/RX11 Floppy Disk System.

Table 6-1
Recommended Tools and Test Equipment

Equipment	Manufacturer and Model/Part No.
Multimeter	Triplett 310 or Simpson 360
Oscilloscope	Tektronix 453
Oscilloscope Probes, Voltage (X10, two required)	Tektronix P6010
Field Service Tool Kit	DEC 29-18303
Head Cleaning Kit (includes TEX pads and wand)	DEC 22-00007 DEC 29-19557 DEC 29-19558
RX8/RX11 Service Kit	
DEC-O-LOG	DEC ECO log and computer on-line synopsis

6.2 CUSTOMER CARE

Although there is no scheduled preventive maintenance, there are two tasks that should be performed on an as-needed basis.

1. Clean the exterior of the RX01 with a damp cloth, using either a solution of nonabrasive cleaner or mild soap.

2. Examine the air filter (Figure 6-1) and clean the element as necessary. Use water and a mild soap, drying thoroughly before reinstalling.

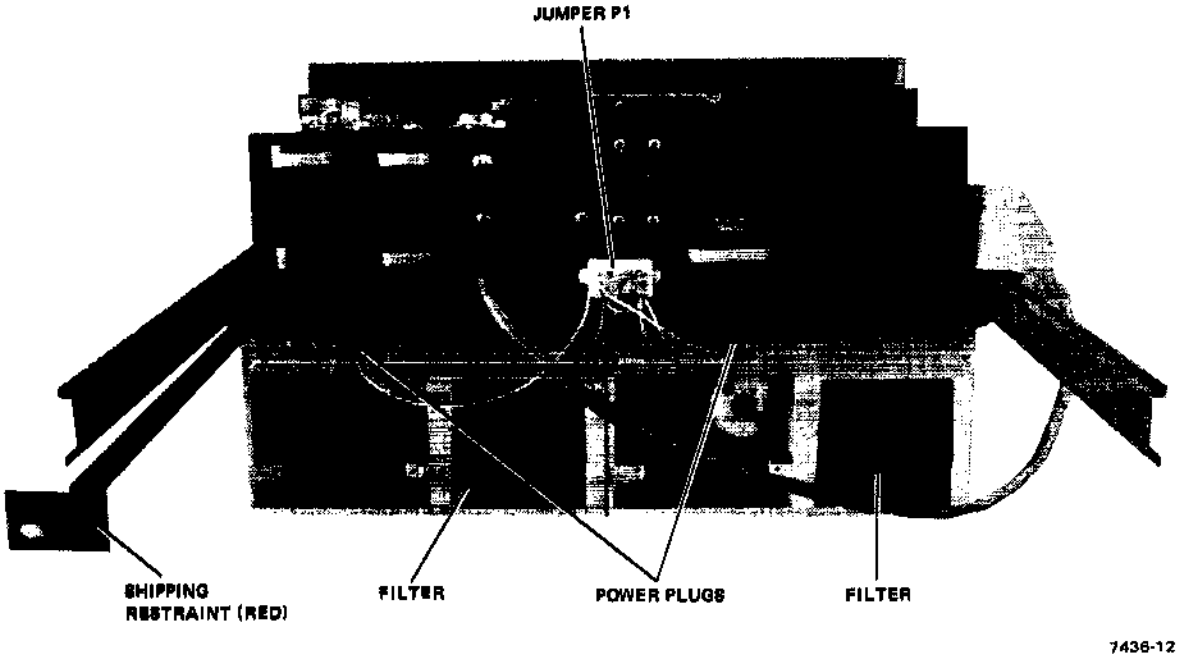


Figure 6-1 RX01, Rear View

6.3 REMOVAL AND REPLACEMENT

The following steps define the procedures for replacing the subassemblies of the RX8/RX11 Floppy Disk System.

6.3.1 Module Replacement

Floppy Disk Controller, M7726 (Figure 6-6)

1. Remove power from the RX01.
2. Unscrew the two captive screws on the right side of the module and raise the module to the servicing position.
3. Remove the plugs in connectors J1, J2, and J4.
4. Lower the module and remove the three screws holding the module onto the hinge.
5. Remove the module and remove the two captive screws.
6. To install a module, insert the two captive screws removed from the original module and perform the reverse of the steps 1–5.

Read/Write Control, M7727 (Figure 6-6)

1. Remove power from the RX01.
2. Raise the floppy disk control module to the servicing position.
3. Remove the plugs from connectors on the module, ensuring that they do not drop into the drive.
4. Remove the six screws holding the module to the frame and remove the module.
5. To install a M7727, replace the screws and plugs removed in steps 3 and 4, ensuring that they are reinstalled into the correct connector (Table 6-2).

Table 6-2
M7727 Connectors

Connector	Description
J1	Disk drive interface cable
J2	Power from H771 power supply
DK0(P3)	Head cable
DK0(P4)	Stepper motor
DK0(P5)	Head load solenoid
DK0(P6)	Index signal
DK0(P7)	Track 0 signal
DK0(P8)	Write protect*
DK1(P3)	Head cable
DK1(P4)	Stepper motor
DK1(P5)	Head load solenoid
DK1(P6)	Index signal
DK1(P7)	Track 0 signal
DK1(P8)	Write protect*

*Not used.

H771 Power Supply Regulator, 70-10718 (Figure 6-6)

1. With the power off, remove the plug from the regulator.
2. Unscrew the leads going to the capacitors, checking with the H771 prints to ensure that the wiring matches the prints.
3. Remove the plugs from the M7726 and M7727.
4. Remove the six screws holding the regulator to the power supply chassis.
5. Replace the regulator by performing the reverse of steps 1—4.

6.3.2 Drive Placement (Figure 6-6)

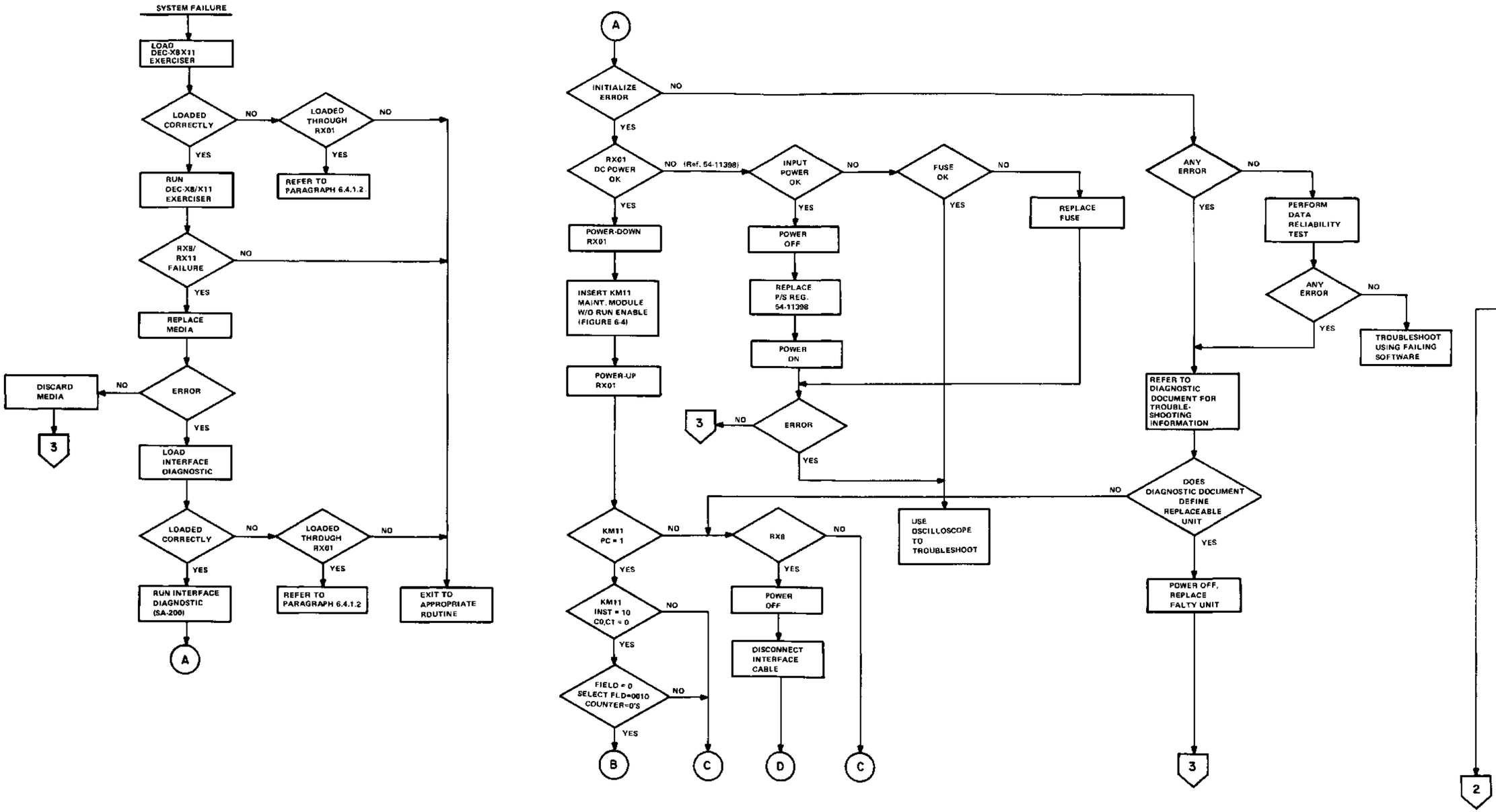
1. With the power removed, remove the power plug in the rear of the drive (Figure 6-1).
2. Remove the plugs for this drive (Table 6-2).
3. Loosen the six screws holding the drive to the chassis.
4. While holding the drive, remove the screws from the four corners.
5. Carefully remove the two remaining screws without allowing the drive to drop down.
6. Slowly lower the drive, guiding the wiring as the drive is lowered.
7. To install a drive, place the two center screws in the holes in the chassis.
8. Raise the drive, guiding the wiring through the hole.
9. With the drive centered, start the two screws carefully so as not to cross-thread them. Do not tighten these screws all the way.
10. Start the remaining screws, being careful not to cross-thread them.
11. Tighten all screws.
12. Insert the plugs listed in Table 6-2.
13. Insert the power plug.

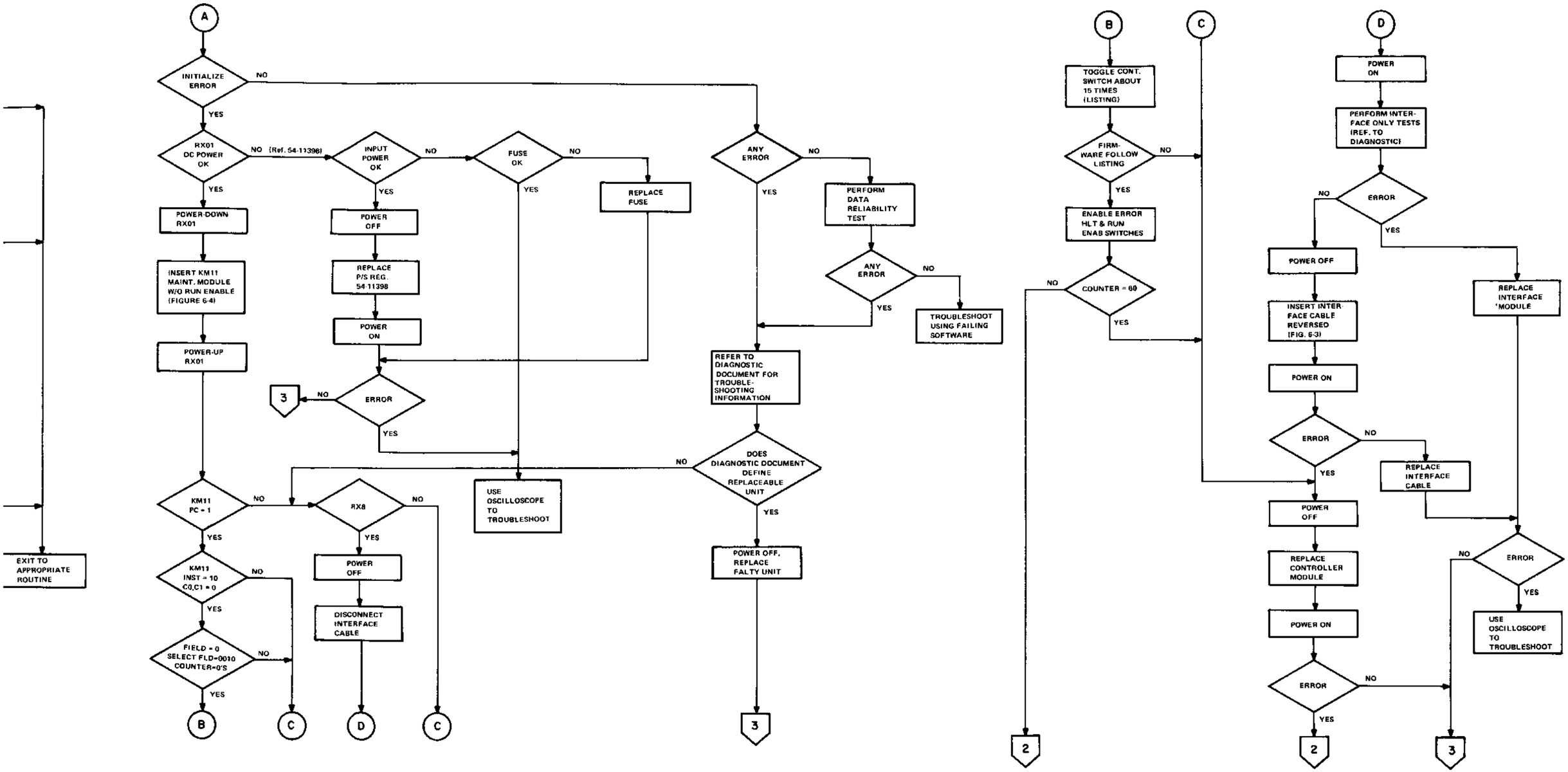
6.4 CORRECTIVE MAINTENANCE

Figure 6-2, Sheet 1, illustrates the method to be used when correcting a fault in the RX8/RX11 system. The proper use of the KM11 module is described in Paragraph 6.4.2.

6.4.1 Errors

6.4.1.1 Interface Diagnostic in Memory – Use Figure 6-2, Sheet 2.


SYSTEM FAILURE
LOAD DEC-X8/X11 EXERCISER
LOADED CORRECTLY
NO
LOADED THROUGH RX01
NO
YES
REFER TO PARAGRAPH 6.4.1.2
YES
RUN DEC-X8/X11 EXERCISER
RX8/ RX11 FAILURE
NO
YES
REPLACE MEDIA
DISCARD MEDIA
NO
ERROR
3
YES
LOAD INTERFACE DIAGNOSTIC
LOADED CORRECTLY
NO
LOADED THRDUGH RX01
NO
YES
REFER TO PARAGRAPH 6.4.1.2
YES
RUN INTERFACE DIAGNOSTIC (SA-200)
EXIT TO APPROPRIATE RDUTINE
A
A
INITIALIZE ERRDR
NO
YES
RX01 DC POWER OK
NO (Ref. 54-11398)
INPUT POWER OK
NO
FUSE OK
NO
YES
YES
YES
REPLACE FUSE
POWER-DOWN RX01
POWER OFF
INSERT KM11 MAINT. MODULE W/O RUN ENABLE (FIGURE 6-4)
REPLACE P/S REG. 54-11398
POWER-UP RX01
POWER ON
3
NO
ERROR
YES
KM11 PC = 1
NO
RX8
NO
YES
YES
USE OSCILLOSCOPE TO TROUBLESHOOT
KM11 INST = 10 CO,C1 = 0
NO
YES
POWER OFF
DISCONNECT INTERFACE CABLE
FIELD = 0 SELECT FLD=0010 COUNTER=0'S
NO
YES
B
C
D
C
ANY ERROR
NO
YES
PERFORM DATA RELIABILITY TEST
ANY ERROR
NO
YES
TROUBLESHOOT USING FAILING SOFTWARE
REFER TO DIAGNOSTIC DOCUMENT FOR TROUBLE-SHOOTING INFORMATION
DOES DIAGNOSTIC DOCUMENT DEFINE REPLACEABLE UNIT
NO
YES
POWER OFF, REPLACE FALTY UNIT
3
B
TOGGLE CONT. SWITCH ABOUT 15 TIMES (LISTING)
FIRM-WARE FOLLOW LISTING
NO
YES
ENABLE ERROR HLT & RUN ENAB SWITCHES
NO
COUNTER > 60
YES
2



CP-1542s

Figure 6-2 Troubleshooting Flow (Sheet 1 of 2)

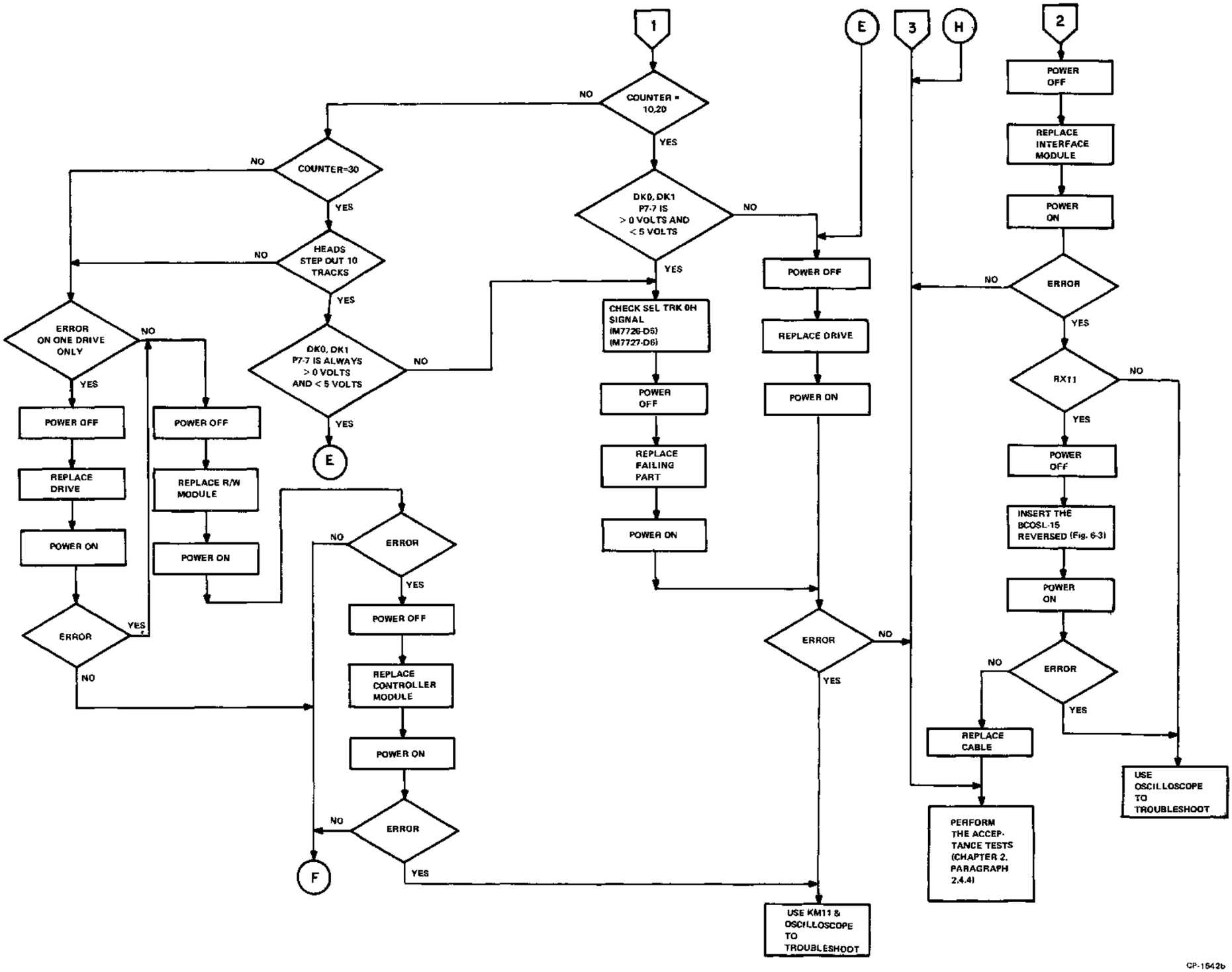


Figure 6-2 Troubleshooting Flow (Sheet 2 of 2)

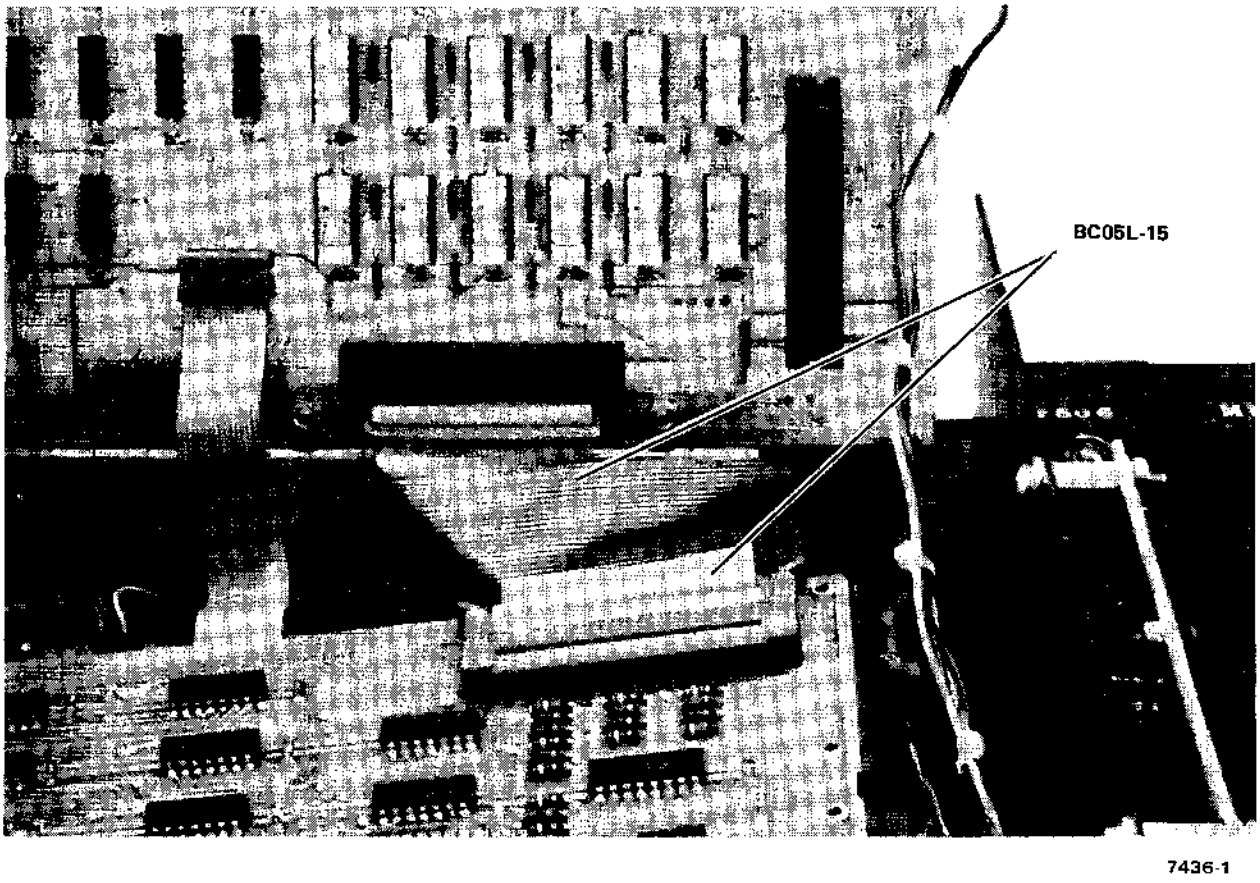


Figure 6-3 BC05L-15 Cable (Reversed) (Sheet 1 of 2)

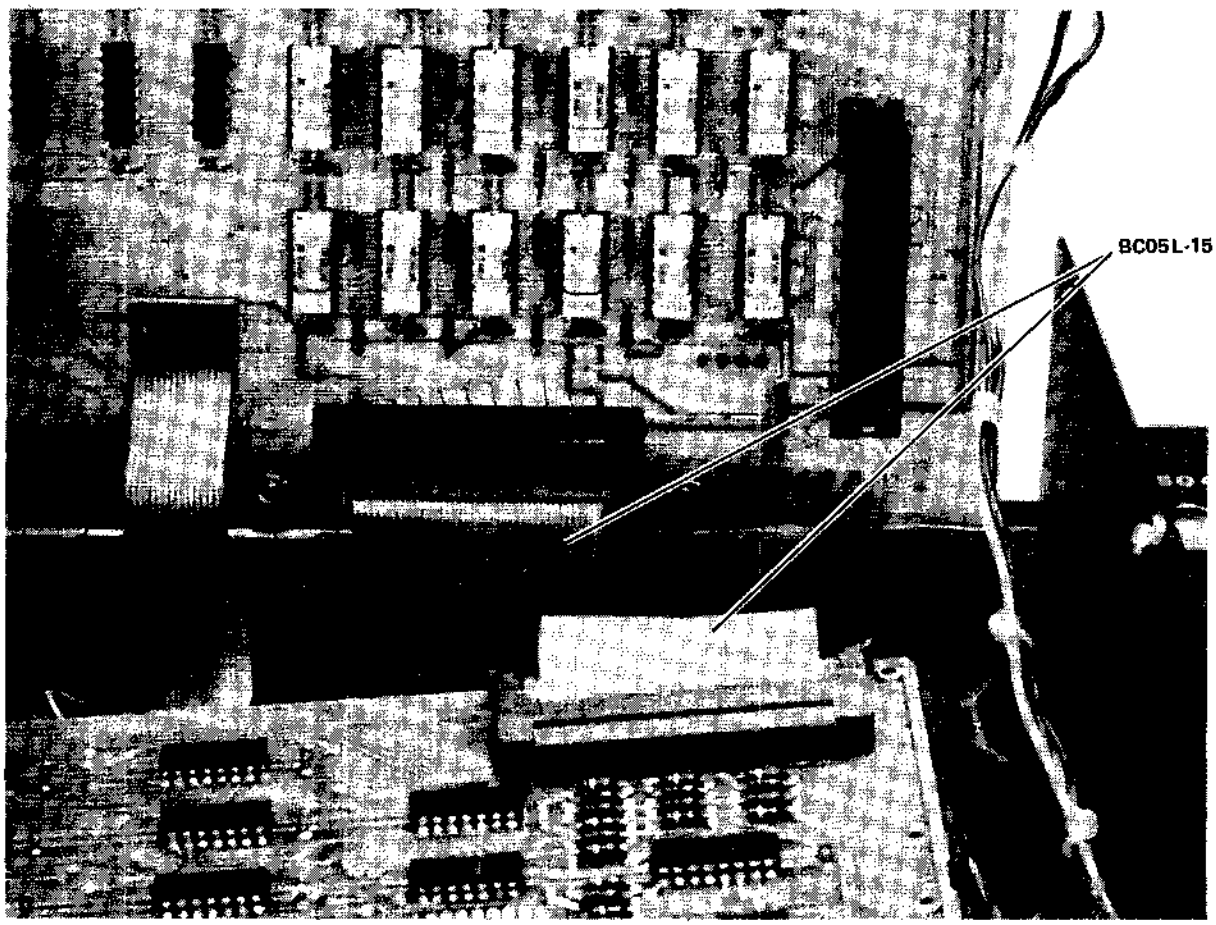


Figure 6-3 BC05L-15 Cable (Correctly Inserted) (Sheet 2 of 2)

6.4.1.2 Diagnostics Not in Memory – Since the RX8/RX11 may be the only input device for a system, there may not be a way to input the diagnostics into the system. In that event, the following routines (Figures 6-4 and 6-5) and the use of the Initialize Diagnostic Routine residing in the controller's firmware may aid in the repair of the floppy disk system.

1. Load the following routines (Figures 6-4 and 6-5) into main memory.

2. Starting at location 200 (RX8) or 1000 (RX11), initiate the program.

3. Examine the status locations for failure information.

   ESTAT,R1
   DSTAT,R0
   EREG,R2
   DRSTAT

4. A good pass with a media installed in drive 0 will be:

   ESTAT/0s
   DSTAT/4_8 or 104_8
   EREG/0s
   DRSTAT/204_8 or 304_8

5. Neither the read/write controller module nor the drives will cause the program to continuously loop on the first check of the Done flip-flop.

6. If the program halts at any location other than the halt at the end of the program, the controller or interface module could be at fault.

7. If the program halts at the end of the program with the following status, the controller is most likely at fault.

   ESTAT/4_8
   DSTAT/0s
   EREG/60_8
   DRSTAT/X

8. All other valid error status will probably be caused by the read/write controller module, the drive, or the floppy disk controller module. It should be noted that a Read Sector is not performed on drive 1; therefore, it is possible for a fault to inhibit reading on both drives without reporting that information.

6.4.2 KM11 Usage

The KM11 maintenance module may be used to single-step through a routine in the floppy controller's firmware. It should be noted that at times the controller will be accessing a signal produced from the media; in this case, the KM11 cannot single-step the microprogram. For the correct method of inserting the KM11, refer to Figure 6-6. The representation of the lights and use of the switches is shown in Figure 6-7. To start a functional routine, the command must be issued from the central processor.

```
                    /RX8 STATUS ROUTINE
          6771              LCD=6771
          6772              XDR=6772
          6774              SER=6774
          6775              SDN=6775
          6777              INIT=6777
          0200              *0200
0200      6775              SDN               /SKIP ON DONE FLAG
0201      5200              JMP .-1           /WAIT FOR FLAG
0202      6774              SER               /SKIP ON ERROR FLAG
0203      5207              JMP .+4           /BRANCH ON NO ERROR
0204      6772              XDR               /TRANSFER DATA - RX01 STATUS TO AC
0205      3227              DCA ERSTAT        /SAVE IN LOCATION ERROR STATUS
0206      5211              JMP .+3           /BRANCH OVER THIS HASH
0207      6772              XDR               /TRANSFER DATA - RX01 STATUS TO AC
0210      3226              DCA DNSTAT        /SAVE IN LOCATION DONE STATUS
0211      1225              TAD RDEREG        /GET READ ERROR REGISTER COMMAND
0212      6771              LCD               /LOAD THE COMMAND REGISTER
0213      6775              SDN               /SKIP ON DONE FLAG
0214      5213              JMP .-1           /WAIT FOR DONE FLAG
0215      6774              SER               /SKIP ON ERROR
0216      7410              SKP               /UNCONDITIONAL SKIP
0217      7402              HLT               /FATAL ERROR - FAILURE TO EXECUTE
                                              /A "READ ERROR REG" COMMAND
0220      6772              XDR               /TRANSFER DATA - RX01 ERROR REG TO AC
0221      3230              DCA EREG          /SAVE IN LOCATION ERROR REG
0222      7402              HLT               /REPLACE WITH NOP (7000) TO LOOP
0223      6777              INIT              /INITIALIZE RX01
0224      5200              JMP 200           /LOOP
0225      0016    RDEREG,   0016
0226      0000    DNSTAT,   0
0227      0000    ERSTAT,   0
0230      0000    EREG,     0
                            $
```

Figure 6-4 RX8 Status Routine

```
                                ;RX11 STATUS ROUTINE
                                        ;R0 = STATUS REG IF DONE COMES UP & NO ERROR FLAG
                                        ;R1 = STATUS REG IF DONE COMES UP & ERROR FLAG
                                        ;R2 = RX01 ERROR REGISTER CONTENTS (SPECIFIC ERROR CODE)
        177170                          RXCS=177170
        177172                          RXDB=177172
        000000                          R0=%0
        000001                          R1=%1
        000002                          R2=%2
        001000                          .=1000
001000  032767  000040  176162  START:  BIT #40,RXCS      ;TEST DONE BIT
001006  001774                          BEQ START         ;WAIT FOR DONE
001010  005767  176154                  TST RXCS          ;TEST ERROR BIT (MSB)
001014  100424                          BMI INITER        ;BRANCH TO INITER ON INITIALIZE ERROR
001016  010067  176150                  MOV R0,RXDB       ;PUT DONE STATUS IN R0
001022  012767  000017  176140  READ:   MOV #17,RXCS      ;ISSUE READ ERROR REG COMMAND & GO BIT
001030  032767  000040  176132  READ1:  BIT #40,RXCS      ;TEST DONE BIT
001036  001774                          BEQ READ1         ;WAIT FOR DONE
001040  005767  176124                  TST RXCS          ;TEST ERROR BIT (MSB)
001044  100001                          BPL READ2         ;BRANCH TO READ2 IF NO ERROR
001046  000000                          HALT              ;FATAL ERROR - ERROR OCCURED
                                                          ;IN "READ ERROR REG" COMMAND
001050  016702  176116          READ2:  MOV RXDB,R2       ;PUT SPECIFIC ERROR CODE IN R2
001054  000000                          HALT              ;NORMAL HALT - REPLACE WITH NOP (240) TO LOOP

001056  012767  040001  176104          MOV #40001,RXCS ;ISSUE RX01 INITIALIZE & GO BIT
001064  000745                          BR START          ;START OVER
001066  016701  176100          INITER: MOV RXDB,R1       ;PUT ERROS STATUS IN R1
001072  000753                          BR READ           ;GO READ ERROR REGISTER
        000001                          .END
```

Figure 6-5 RX11 Status Routine

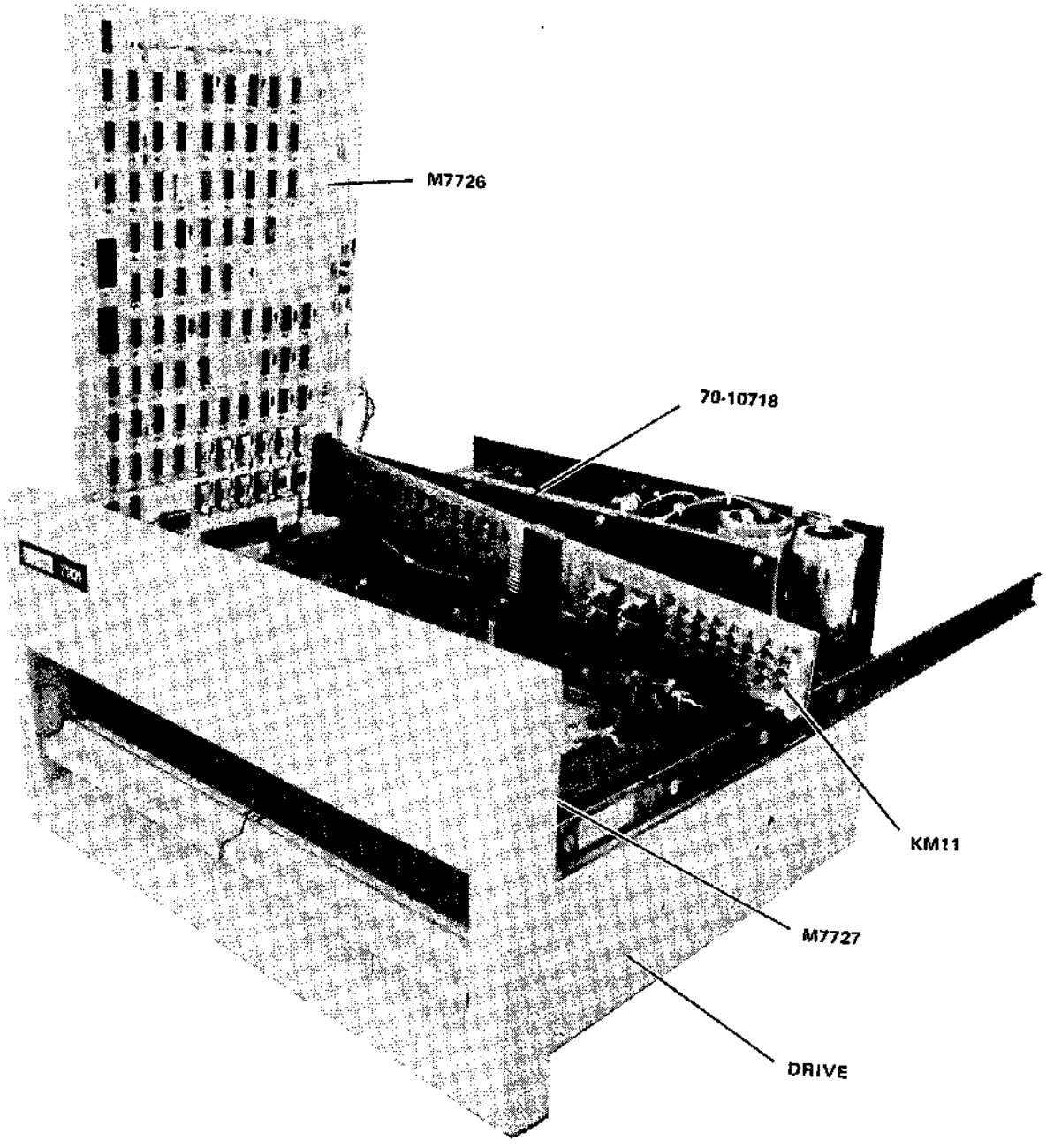
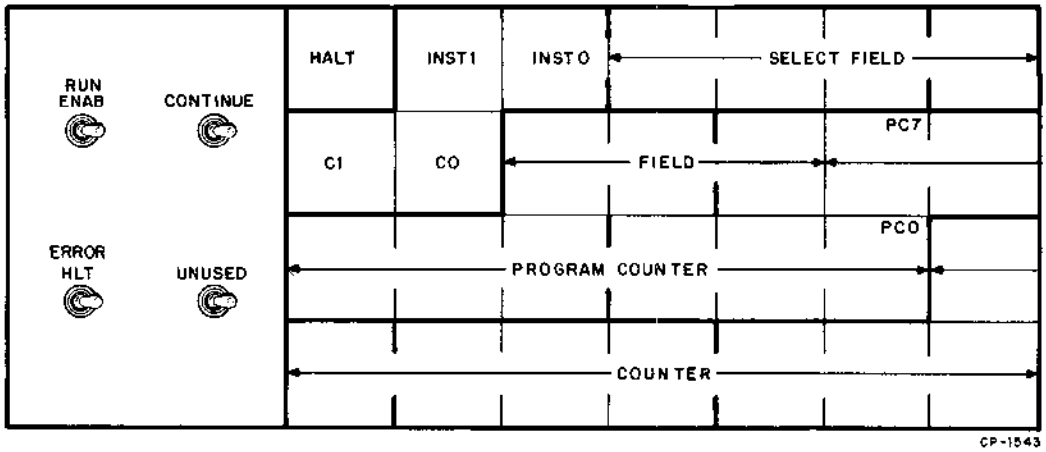


Figure 6-6 KM11 Maintenance Module Inserted



Switch	Function
RUN ENAB	ON: M7726 Clock OFF: Maint. Clock Pulses (Continue)
CONTINUE	ON: Advance Controller Firmware Once OFF: –
ERROR HLT	ON: Halt Controller When Error Detected OFF: Do Not Halt On Error Condition

Lights	Function
HALT	Firmware Halted
INST 1 and 0	Instruction Bits
SELECT FLD	Selects 1 of 16 Conditions (Depends on Instruction)
C0 and C1	Control Functions (Depends on Instruction)
FIELD	ROM Field
PROGRAM COUNTER (0–7)	(Halted) Address +1 of Instruction Displayed
COUNTER (0–7)	Displays Contents of Counter Register

Figure 6-7 KM11 Light and Switch Definitions for RX01

Table 6-3
Power Supply Output Voltages

Voltage	Tolerance	Measured At
+5 Vdc	⩾ +4.75 Vdc ⩽ +5.25 Vdc Ripple ⩽ 200 mV (p-p)	P1-4 P2-4
+9.5 Vdc	⩾ +9.0 Vdc ⩽ +10.3 Vdc Ripple ⩽ 2.0 V (p-p)	P2-7
+24 Vdc	⩾ +23.6 Vdc ⩽ +28.0 Vdc Ripple ⩽ 1.2 V (p-p)	P1-1
-5 Vdc	⩾ -4.6 Vdc ⩽ -5.6 Vdc Ripple ⩽ 200 mV (p-p)	P1-6
+10 Vac		J1-1, 3
+24 Vac		J1-2, 4

NOTE
This table should be used in conjunction with the DC Power Checks performed with Figure 6-2, Sheet 1.

Reader's Comments

RX8/RX11 FLOPPY DISK SYSTEM
MAINTENANCE MANUAL
EK-RX01-MM-002

Your comments and suggestions will help us in our continuous effort to improve the quality and usefulness of our publications.

What is your general reaction to this manual? In your judgment is it complete, accurate, well organized, well written, etc.? Is it easy to use? _____

_____

_____

_____

What features are most useful? _____

_____

_____

_____

What faults do you find with the manual? _____

_____

_____

_____

Does this manual satisfy the need you think it was intended to satisfy? _____

Does it satisfy *your* needs? _____ Why? _____

_____

_____

_____

Would you please indicate any factual errors you have found. _____

_____

_____

_____

Please describe your position. _____

Name _____ Organization _____

Street _____ Department _____

City _____ State _____ Zip or Country _____

CUT OUT ON DOTTED LINE

Fold Here

Do Not Tear - Fold Here and Staple

FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:

Digital Equipment Corporation
Technical Documentation Department
146 Main Street
Maynard, Massachusetts 01754