EK-RX02-UG-001

# RX02 Floppy Disk System User's Guide

digital equipment corporation • maynard, massachusetts

1st Edition, July 1978

The material in this manual is for informational purposes and is subject to change without notice.

Digital Equipment Corporation assumes no responsibility for any errors which may appear in this manual.

Printed in U.S.A.

**This document was set on DIGITAL's DECset-8000 computerized typesetting system.**

11/82-15

# CONTENTS

Page

# CONTENTS (Cont)

Page

# CONTENTS (Cont)

Page

# CONTENTS (Cont)

# FIGURES

# FIGURES (Cont)

# TABLES

# PREFACE

This manual is intended to provide the user with sufficient information to correctly set up and operate the RX02 Floppy Disk System in any of the various configurations that are available for use with the PDP-8, PDP-11, or LSI-11 computers. The manual presents general, installation, user, and programming information for the RX02 Floppy Disk System and the interface options associated with the PDP-8, PDP-11, and LSI-11 computer systems.

# CHAPTER 1
# GENERAL INFORMATION

## 1.1 INTRODUCTION

The RX02 is a low cost, random access mass memory device that stores data in fixed length blocks on flexible diskettes with preformatted industry standard headers. The RX02 interfaces with either a PDP-8, a PDP-11, or an LSI-11 system. Various interface modules are selected according to the computer being used and either single or double density recording. The various configurations are:

| Designation | Computer | Interface Module | Recording Density |
|---|---|---|---|
| RX8E | PDP-8 | M8357 | Single |
| RX28 | PDP-8 | M8357 | Single or Double |
| RX11 | PDP-11 | M7846 | Single |
| RX211 | PDP-11 | M8256 | Single or Double |
| RXV11 | LSI-11 | M7946 | Single |
| RXV21 | LSI-11 | M8029 | Single or Double |

**NOTE**

**The single density recording configurations RX8E, RX11, and RXV11 are compatible with the RX01 Floppy Disk System when the M7744 controller module has been switched to be compatible with these configurations. (See Table 2-2.)**

The RX02 consists of 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, a PDP-11 interface module, or an LSI-11 interface module. The amount of data that can be stored on the RX02 varies according to the configuration. The recording density can be different for each drive. For each drive system using double density recording, up to 512K 8-bit bytes of data (PDP-8, PDP-11, LSI-11) or 256K 12-bit words (PDP-8) can be stored and retrieved. For each drive system using single density recording, up to 256K 8-bit bytes of data or 128K 12-bit words (PDP-8) can be stored and retrieved. The RX02 interfaces with IBM-compatible devices when single density data recording is used.

For single or double density recording, the RX02 is used with either an M8357 interface module (PDP-8), an M8256 interface module (PDP-11), or an M8029 interface module (LSI-11). The interface modules convert the RX02 I/O bus to the bus structure of the computer being used. Each module controls the interrupts to the CPU initiated by the RX02 and handles the data interchange between the RX02 and the host computer. Each interface module is powered by the host processor.

In addition, the RX02 is used for single density recording when it is configured to be compatible with the RX01. The interface module used is either an M8357 (PDP-8), an M7846 (PDP-11), or an M7946 (LSI-11).

To record or retrieve data the RX02 performs implied seeks. Given an absolute sector address, the RX02 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.

## 1.2 GENERAL DESCRIPTION

An RX02 Floppy Disk System consists of the following components:

M7744 Controller Module
M7745 Read/Write Electronics Module
H771-A, -C, or -D Power Supply
RX02-CA Floppy Disk Drive (60 Hz max of 2)
RX02-CC Floppy Disk Drive (50 Hz max of 2)

One interface module is used:

M8357 (PDP-8, Programmed I/O)
M7846 (PDP-11, Programmed I/O)
M7946 (LSI-11, Programmed I/O)
M8256 (PDP-11 with DMA)
M8029 (LSI-11 with DMA)

All components except the interface modules are housed in a 10-1/2 inch rack-mountable box. The power supply, M7744 module, and M7745 module are mounted above the drives. Interconnection from the RX02 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: part A shows the configuration for a bus interface with DMA; part B shows the configuration for all Omnibus interfaces (programmed I/O); part C shows the configuration for a bus interface (programmed I/O) that is RX01 compatible. Figure 1-2 is a front view of a dual drive system.

### 1.2.1 Interface Modules

The interface modules plug into a slot on the bus for PDP-8, PDP-11, and LSI-11 computers. Figure 1-3 shows the outline of the various modules and areas of interest on each module.

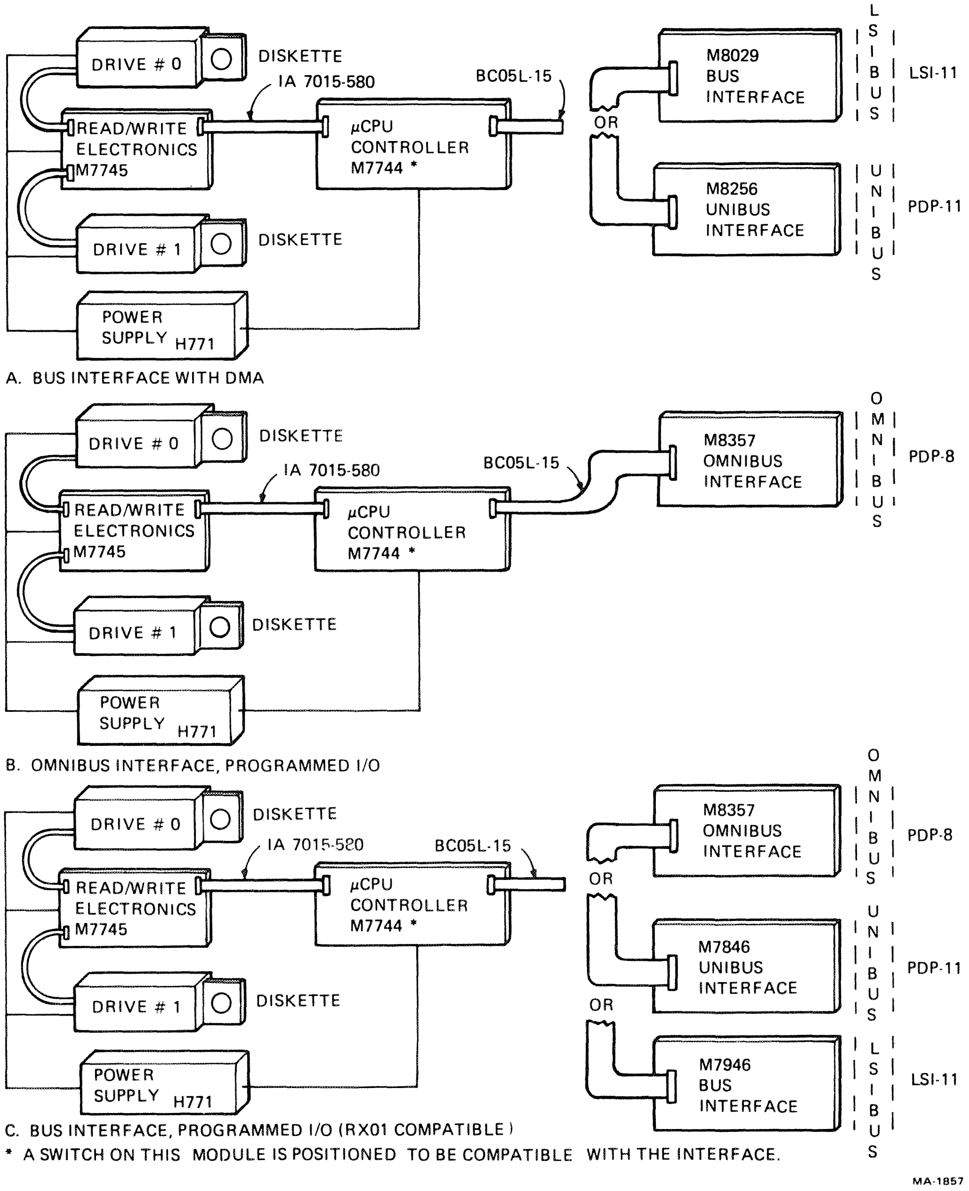


Figure 1-1 Floppy Disk Configuration

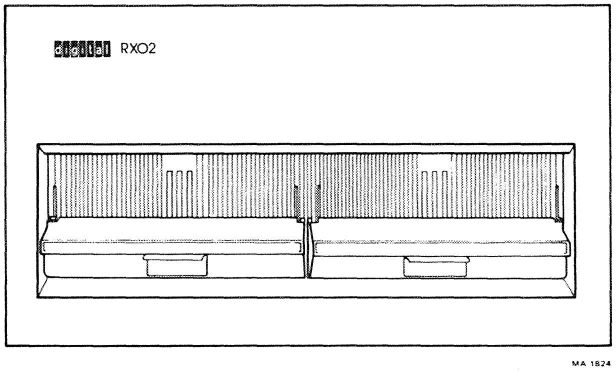


Figure 1-2 Front View of the Floppy Disk System

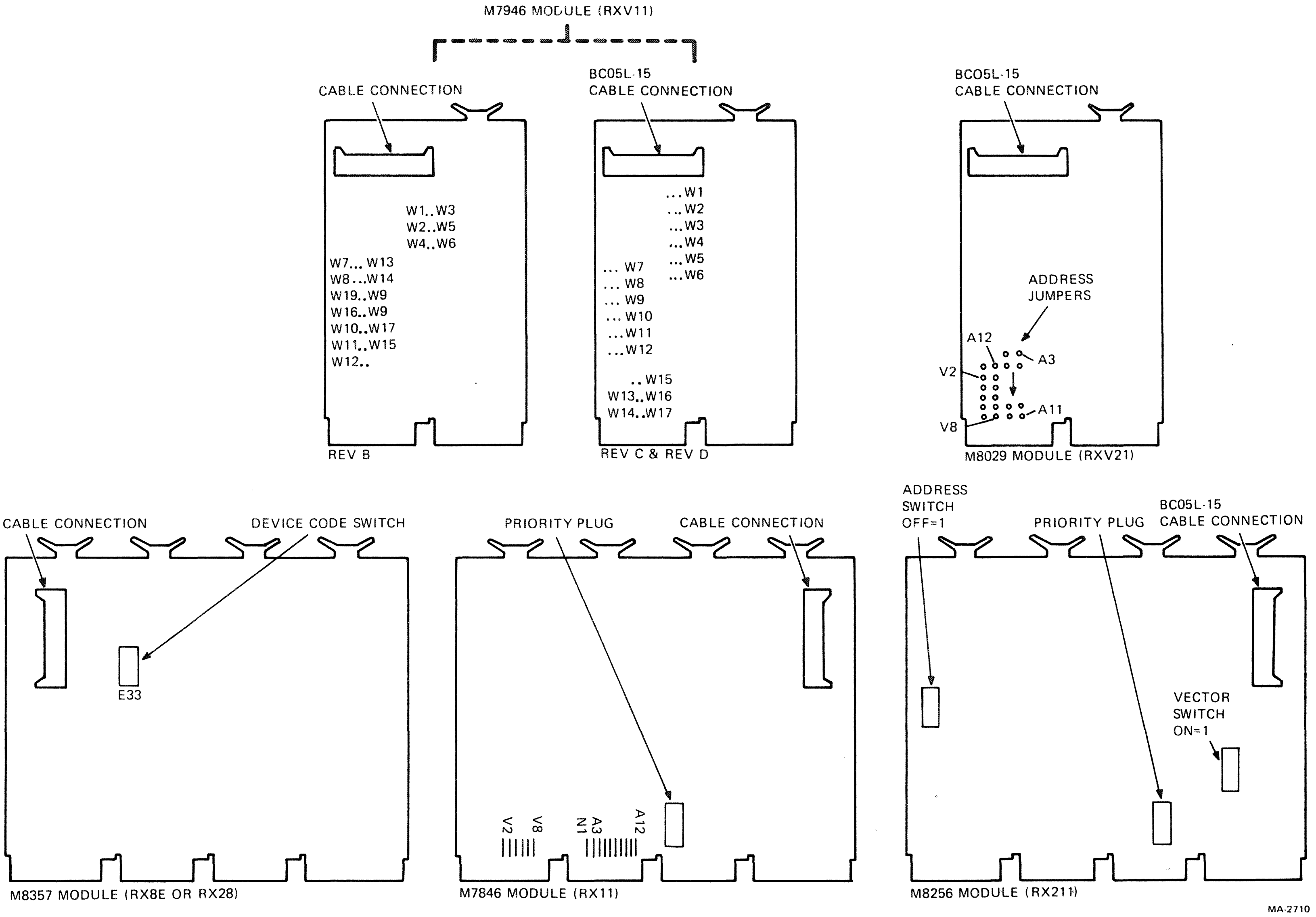


Figure 1-3 Interface Modules

### 1.2.2 Microprogrammed Controller

The M7744 microprogrammed controller module is located in the RX02 cabinet as shown in Figure 1-4. The M7744 is hinged on the left side and lifts up for access to the M7745 read/write electronics module.

### 1.2.3 Read/Write Electronics

The M7745 read/write electronics module is located in the RX02 cabinet as shown in Figure 1-4.

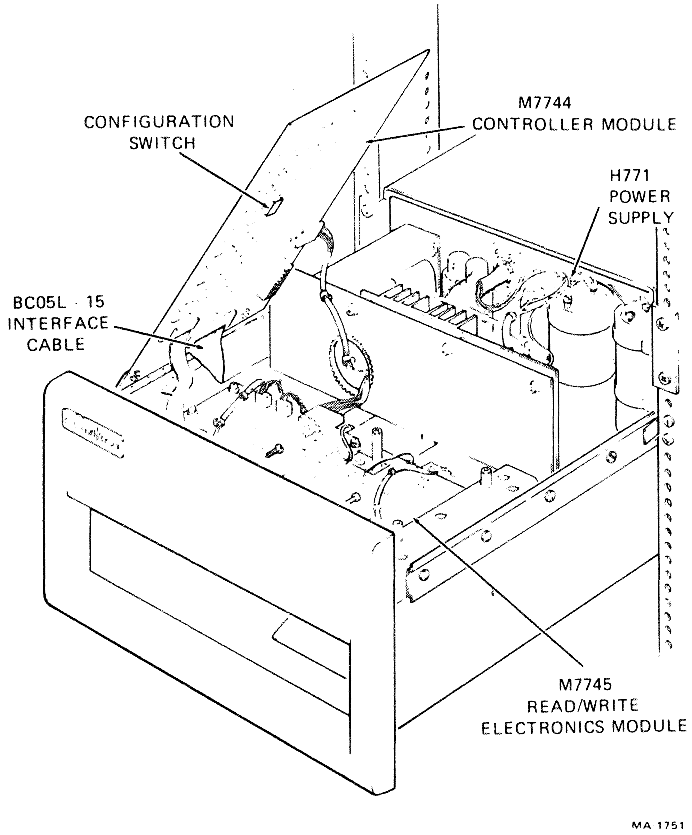


Figure 1-4 Top View of RX02

### 1.2.4 Electromechanical Drive

A maximum of two drives can be attached to the read/write electronics. The electromechanical drives are mounted side by side under the read/write electronics board (M7745). Figure 1-5 is an underside view of the drive showing the drive motor connected to the spindle by a belt. (This belt and the drive pulley are different on the 50 Hz and 60 Hz units; see Paragraph 2.1.3.2 for complete input power modification requirements.)

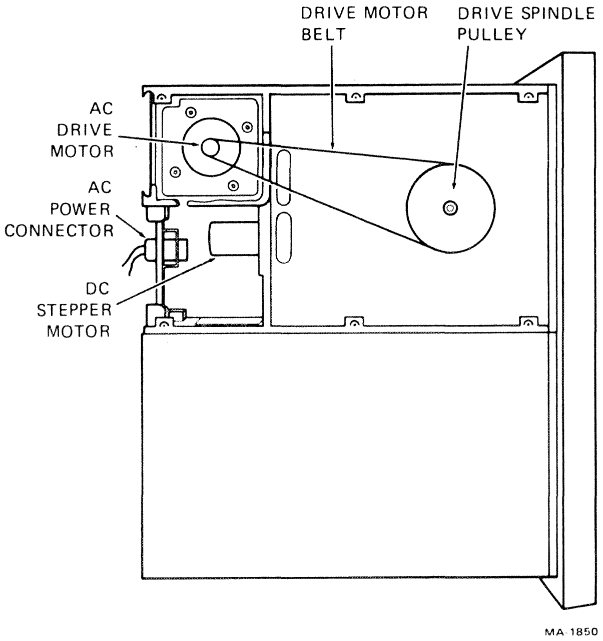


Figure 1-5 Underside View of Drive

### 1.2.5 Power Supply

The H771 power supply is mounted at the rear of the RX02 cabinet as shown in Figure 1-4. The H771-A is rated at 60 Hz ±1/2 Hz over a voltage range of 90–128 Vac. The H771-C and -D are rated at 50 Hz + 1/2 Hz over four voltage ranges:

90–120 Vac<br>
100–128 Vac } 3.5 A circuit breaker; H771-C

184–240 Vac<br>
200–256 Vac } 1.75 A circuit breaker; H771-D

Two configuration plugs are provided to adapt the H771-C or -D to each voltage range. This is not applicable to the H771-A.

## 1.3 OPTION DESCRIPTION

The optional interface modules that are used to interface the RX02 with a PDP-8, PDP-11, and LSI-11 are listed in Paragraphs 1.1 and 1.2. (Each module is powered by the host processor.) The module selected is determined by the computer being used and whether the data interchange is between either IBM system 3740 compatible devices or DIGITAL system double density devices. Also, when an M7744 controller module's configuration switch is set to be compatible, the RX02 can operate as an RX01. The RX02 interfaces with IBM compatible devices when single density data recording is used. The RX02 interfaces with DIGITAL system double density recording devices when the controller module configuration switch is positioned to be compatible with RX28, RX211, and RXV21 configurations.

### 1.3.1 Operation For Single Density Recording Only (RX8E, RX11, RXV11)

**1.3.1.1 PDP-8 Operation** – The RX02 connects to the M8357 Omnibus interface module. This module converts the RX02 I/O bus to PDP-8 family Omnibus structure. It controls interrupts to the CPU initiated by the RX02, controls data interchange between the RX02 and the host CPU by programmed I/O, and handles input/output transfers used for maintenance status conditions.

**1.3.1.2 PDP-11 Operation** – The RX02 connects to the M7846 Unibus interface module. This module converts the RX02 I/O bus to PDP-11 Unibus structure. It controls interrupts to the CPU initiated by the RX02, decodes Unibus addresses for register selection, and handles data interchange between the RX02 and the host CPU main memory by programmed I/O.

**1.3.1.3 LSI-11 Operation** – The RX02 connects to the M7946 LSI-11 bus interface module. This module converts the RX02 I/O bus to the LSI-11 bus structure. It controls interrupts to the CPU initiated by the RX02, decodes LSI-11 bus addresses for register selection, and transfers data between the RX02 and the host CPU main memory by programmed I/O.

### 1.3.2 Operation For Single or Double Density Recording (RX28, RX211, RXV21)

**1.3.2.1 PDP-8 Operation** – The RX02 connects to the M8357 Omnibus interface module. This module converts the RX02 I/O bus to PDP-8 family Omnibus structure. It controls interrupts to the CPU initiated by the RX02, controls transfer of data between the RX02 and host CPU by programmed I/O, and handles input/output transfer used to test status conditions.

**1.3.2.2 PDP-11 Operation** – The RX02 connects to the M8256 Unibus interface module. This module converts the RX02 I/O bus to PDP-11 Unibus structure. It controls interrupts to the CPU initiated by the RX02, decodes Unibus addresses for register selection, and initiates NPR requests to transfer data between the RX02 and the host CPU main memory.

**1.3.2.3 LSI-11 Operation** – The RX02 connects to the M8029 LSI-11 bus interface module. This module converts the RX02 I/O bus to the LSI-11 bus structure. It controls interrupts to the CPU initiated by the RX02, decodes LSI-11 bus addresses for register selection, and initiates NPR requests to transfer data between the RX02 and the host CPU main memory.

## 1.4 SPECIFICATIONS

| | |
|---|---|
| System Reliability | |
| Minimum number of revolutions per track | 3 million/media (head loaded) |
| Seek error rate | 1 in $10^6$ seeks |
| Soft data error rate | 1 in $10^9$ bits read or written |
| Hard data error rate | 1 in $10^{12}$ bits read or written |

**NOTE**

**The above error rates only apply to DEC approved media that is properly cared for. Seek error and soft data 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 10 additional tries or less. "Hard" errors cannot be recovered. Seek error retries should be preceded by a recalibrate.**

Drive Performance

| Capacity | Recording | 8-bit bytes | 12-bit words |
|---|---|---|---|
| Per diskette | FM | 256,256 | 128,128 |
| | MFM | 512,512 | 256,256 |
| Per track | FM | 3,328 | 1,664 |
| | MFM | 6,656 | 3,328 |
| Per sector | FM | 128 | 64 |
| | MFM | 256 | 128 |

| Data transfer rate | |
|---|---|
| Diskette to controller buffer | 4 μs/data bit (FM)<br>2 μs/data bit (MFM) |
| Buffer to CPU interface | 1.2 μs/bit |

**NOTE**

**PDP-8 interface can operate in 8- or 12-bit modes under software control.**

| | |
|---|---|
| Track-to-track move | 6 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 (0–$32_8$) |
| Recording technique | Double frequency (FM) or modified MFM |
| Bit density maximum on inner track | 3200 bpi (FM) or modified (MFM) |
| Track density | 48 tracks/inch |
| Average access | 262 ms, computed as follows: |

$$\overbrace{77 \text{ tks}/3 \times 6 \text{ ms}}^{\text{Seek}} \quad \overbrace{+\ 25 \text{ ms}}^{\text{Settle}} \quad \overbrace{+\ 166 \text{ ms}/2 = 262 \text{ ms}}^{\text{Rotate}}$$

**Environmental Characteristics**

| Temperature | |
|---|---|
| RX02, operating | 15° to 32° C (59° to 90° F) ambient;<br>maximum temperature gradient = 11° C/hr (20° F/hr) |
| RX02, nonoperating | -35° to +60° C (-30° to +140° F) |
| Media, nonoperating | -35° to +52° C (-30° to +125° F) |

**NOTE**

**Media temperature must be within operating temperature range before use.**

| | |
|---|---|
| Heat Dissipation (RX02 System) | Less than 225 Btu/hr |

| Relative humidity | |
|---|---|
| RX02, operating | 25° C (77° F) maximum wet bulb<br>2° C (36° F) minimum dew point<br>20% to 80% relative humidity |

| | |
|---|---|
| RX02, 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 | |
| RX02 | 5 A at +5 Vdc, 25 W; 0.14 A at -5 Vdc, 0.7 W; 1.3 A t +24 Vdc, 31 W |
| PDP-11 interface (M7846, M8256) | 1.8 A at 5 Vdc |
| PDP-8 interface (M8357) | 1.5 A at 5 Vdc |
| LSI-11 interface (M7946, M8029) | 1.8 A at 5 Vdc |
| AC power | 4 A at 115 Vac<br>2 A at 230 Vac |

## 1.5 SYSTEMS COMPATIBILITY

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

### 1.5.1 Media

The media used on the RX02 Floppy Disk system is compatible with the IBM 3740 family of equipment and is shown in Figure 1-6. The "diskette" media was designed by applying tape technology to disk architecture, resulting in a flexible oxide-on-mylar surface. The diskette is 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.

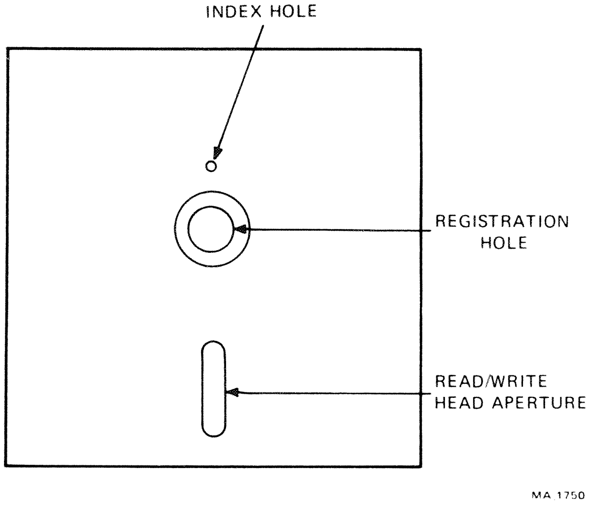


Figure 1-6 Diskette Media

### 1.5.2 Recording Scheme

There are two recording schemes used in the RX02: double frequency (FM) and modified Miller code (MFM). The FM scheme is used for single density data recording which is compatible with IBM system 3740 devices. (When this recording scheme is used and the RX02 is configured as shown in Figure 1-1 part C, the RX02 is compatible with the RX01.) The MFM scheme is used for double density data recording which is compatible with DIGITAL double density devices but is not compatible with other manufacturers.

**1.5.2.1 Double Frequency** (FM) – For the double frequency recording scheme data is recorded between bits of a constant clock stream. The clock stream consists of a continuous pattern of one flux reversal every four μs (Figure 1-7). 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 1-7, would appear as a "2F" bit stream, and a continuous stream of zeros, shown in the top waveform in Figure 1-7, would appear as a "1F" or fundamental frequency bit stream.

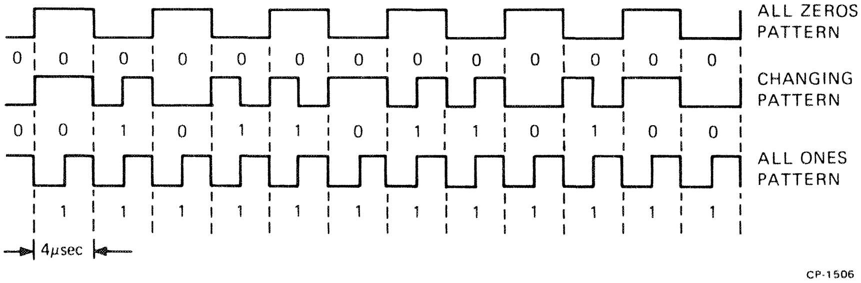


Figure 1-7 Flux Reversal Patterns for FM

**1.5.2.2 Miller Code** (MFM) – MFM or Miller code encodes clocks between data bits of a continuous data stream. The data stream consists of flux reversals for a data "one" and no flux reversal for a data "zero." A clock is recorded only between data "zeros." Because it is possible to have double density data fields map into a preamble and ID mark, the MFM encoding is modified slightly to prevent a false header from being detected within a double density data field.

**NOTE**

**The modified MFM encoding is not compatible with other manufacturers.**

The encoding algorithms for implementing modified MFM are:

Encoding Algorithm #1 (MFM or Miller Code Algorithm)

| | Data | Encoded Data | | |
|---|---|---|---|---|
| **Dn** | **Dn + 1** | **Dn** | **Cn** | **Dn + 1** |
| 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 |

Encoding Algorithm #2 (MFM Modified Algorithm)

| Data | | | | | |
|---|---|---|---|---|---|
| Dn | Dn + 1 | Dn + 2 | Dn + 3 | Dn + 4 | Dn + 5 |
| 0 | 1 | 1 | 1 | 1 | 0 |

| Encoded Data | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Dn | Cn | Dn + 1 | Cn + 1 | Dn + 2 | Cn + 2 | Dn + 3 | Cn + 3 | Dn + 4 | Cn + 4 | Dn + 5 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |

The decoding algorithm used in data separation is:

| Encoded | | | Decoded | |
|---|---|---|---|---|
| Dn | Cn | Dn + 1 | Dn | Dn + 1 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |

Figure 1-8 shows the waveforms that are generated for a data stream of zeros and ones when FM code, MFM code, and modified MFM code are used.

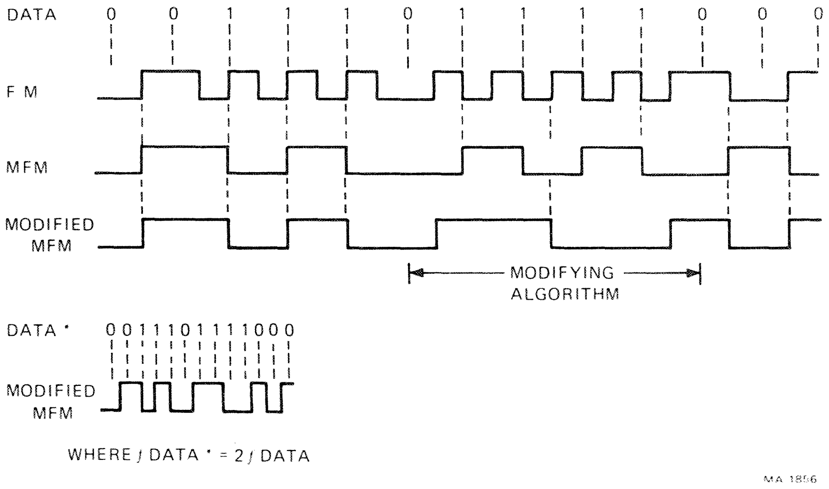


Figure 1-8 FM versus MFM Encoding

### 1.5.3 Logical Format

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 1-9). Each sector contains two major fields: the header field and the data field (Figure 1-10).

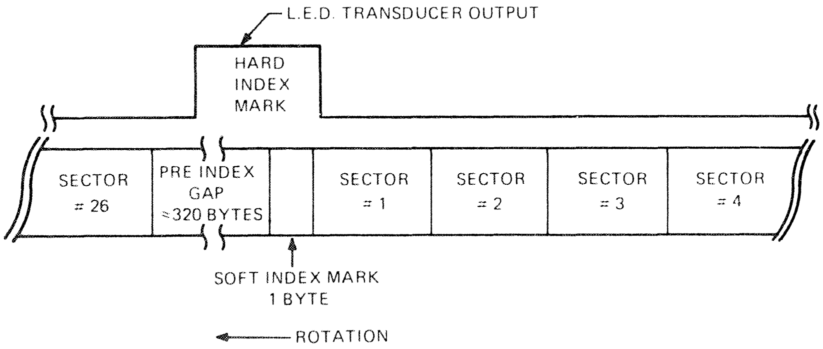


Figure 1-9 Track Format (Each Track)

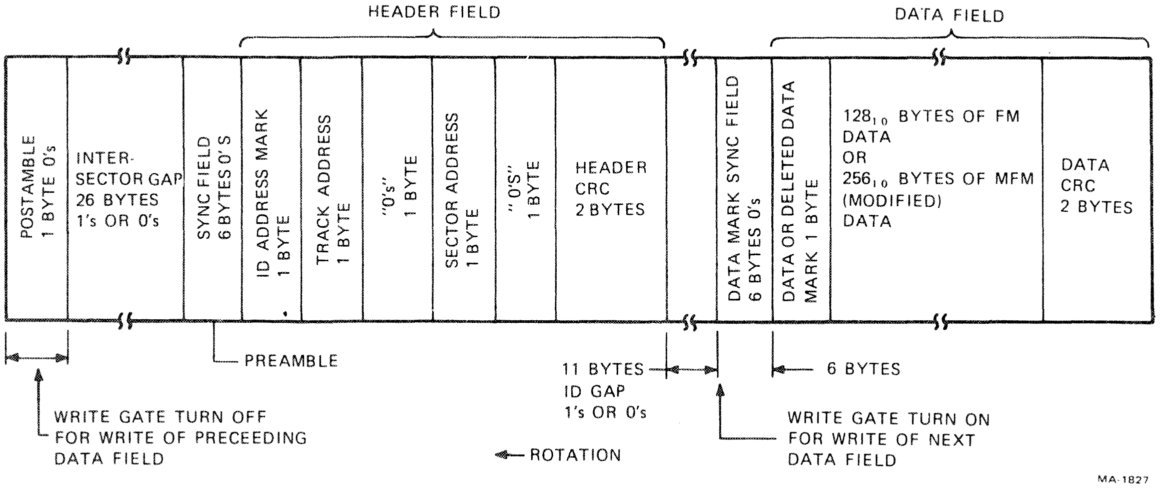

HEADER FIELD
DATA FIELD
POSTAMBLE 1 BYTE 0's
INTER-SECTOR GAP 26 BYTES 1's OR 0's
SYNC FIELD 6 BYTES 0'S
ID ADDRESS MARK 1 BYTE
TRACK ADDRESS 1 BYTE
"0's" 1 BYTE
SECTOR ADDRESS 1 BYTE
"0'S" 1 BYTE
HEADER CRC 2 BYTES
DATA MARK SYNC FIELD 6 BYTES 0's
DATA OR DELETED DATA MARK 1 BYTE
128₁₀ BYTES OF FM DATA OR 256₁₀ BYTES OF MFM (MODIFIED) DATA
DATA CRC 2 BYTES
PREAMBLE
11 BYTES ID GAP 1's OR 0's
6 BYTES
WRITE GATE TURN OFF FOR WRITE OF PRECEEDING DATA FIELD
ROTATION
WRITE GATE TURN ON FOR WRITE OF NEXT DATA FIELD
MA-1827

Figure 1-10 Sector Format (Each Sector)

**1.5.3.1 Header Field Description** – The header field is broken into seven bytes (eight bits/byte) of information and is preceded by a field of at least six bytes of zeros for synchronization. The header and its preamble are always recorded in FM.

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. (Data = FE hex, clock = C7 hex.)

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.

3. Byte No. 3: – Zeros

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. There is no sector 0.

5. Byte No. 5: – Zeros

6,7. 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 the IBM 3740 polynomial.

**1.5.3.2 Data Field Description** – The data field contains either $131_{10}$ or $259_{10}$ bytes of information depending on the recording scheme. This field is preceded by a field of zeros for synchronization and the header field (Figure 1-10).

1. Byte No. 1: Data or Deleted Data Address mark – This byte is always recorded in FM and is unique because it contains missing clocks. It is decoded by the controller to identify the beginning of a data field. The deleted data mark is not used during normal operation but the RX02 can identify and write deleted data marks under program control as required. There is a unique address mark for each density as shown in the following table. One of these marks is the first byte of each data field.

Table 1-1 Data Address Mark Code

| | | Hex Byte | |
|---|---|---|---|
| **Mark** | **Density** | **Data** | **Clock** |
| Data | FM | FB | C7 |
| | MFM mod. | FD | C7 |
| DELETED DATA | FM | F8 | C7 |
| | MFM mod. | F9 | C7 |

2. Bytes No. 2: –129 (FM) or –257 (MFM modified) – This is the data field and it can be recorded in either FM or MFM (modified). It is used to store $128_{10}$ or $256_{10}$ (depending upon encoding) 8-bit bytes of information.

NOTE
**Partial data fields are not recorded.**

3. Bytes No. 130 and 131 or 258 and 259 – These bytes comprise the CRC character that is calculated for each sector from the first 129 or 257 data field bytes using the industry standard polynomial division algorithm designed to detect the types of failures most likely to occur in recording on the floppy media. These bytes will be recorded with the same encoding scheme as the data field.

**1.5.3.3 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 RX02 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.

**1.5.3.4 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 IBM 3740 series equipment.

# CHAPTER 2
# INSTALLATION

This chapter contains information that is required for site preparation, unpacking, installation, and testing of the RX02 Floppy Disk System. Information is also provided to identify the various system configurations that are available.

## 2.1 SITE PREPARATION

### 2.1.1 Space

The RX02 is a cabinet-mountable unit that may be installed in a standard Digital Equipment Corporation cabinet. This rack-mountable version is approximately 28 cm high, (10-1/2 inches), 48 cm wide, (19 inches) and 42 cm deep (16-1/2 inches) as shown in Figure 2-1.

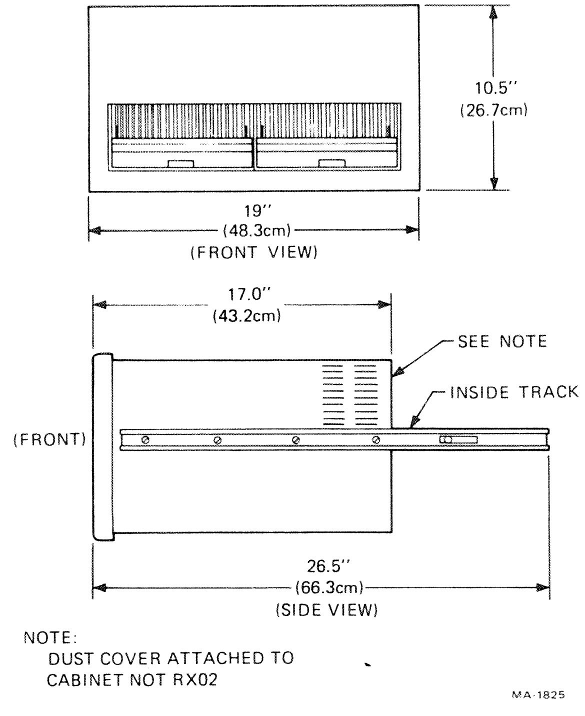


Figure 2-1 RX02 Outline Dimensions

When the RX02 is mounted in a cabinet (Figure 2-2), provision should be made for service clearances of approximately 56 cm (22 inches) at the front and rear of the cabinet so that the RX02 can be extended or the cabinet rear door opened.

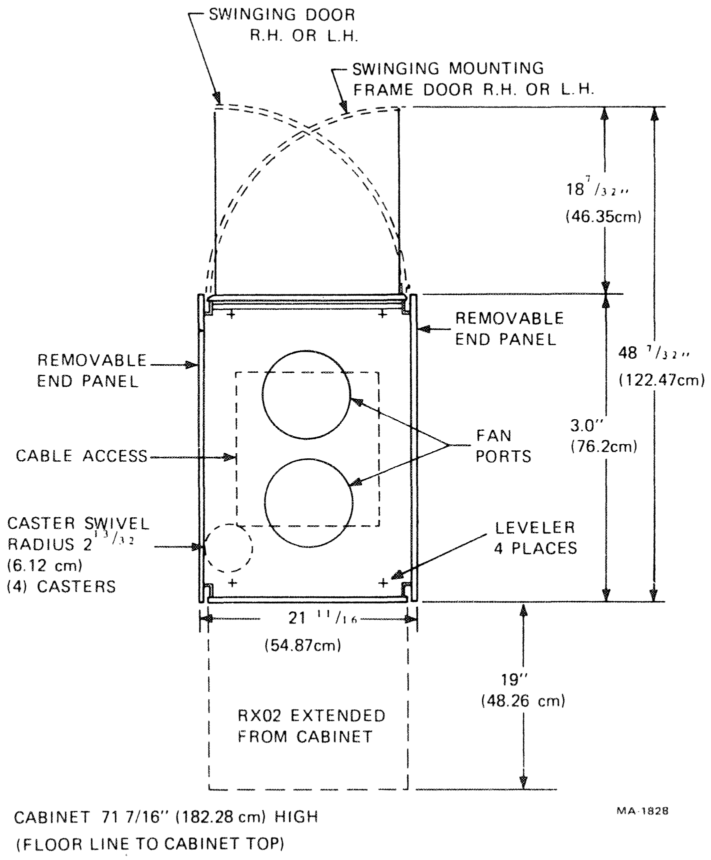


Figure 2-2 Cabinet Layout Dimensions

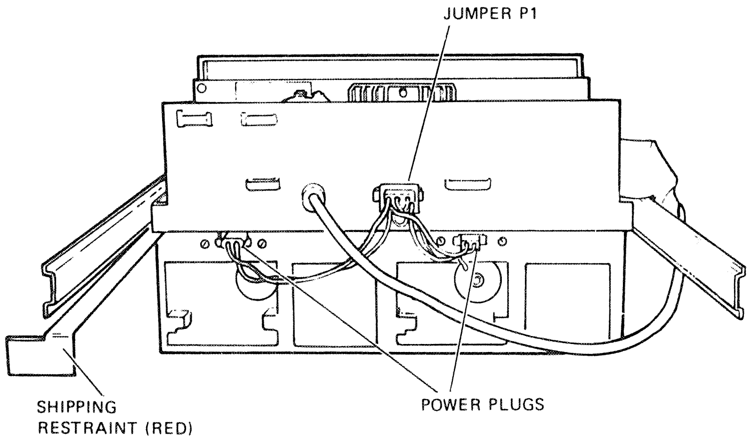
### 2.1.2 Cabling

The standard interface cable provided with an RX02 (BC05L-15) is 4.6 m (15 ft) in length; the positioning of the RX02 in relation to the central processor should be planned to take this into consideration. The RX02 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 is about 2.7 m (9 ft) long.

### 2.1.3 AC Power

**2.1.3.1 Power Requirements** – The RX02 is designed to use either a 60 Hz or a 50 Hz power source. The 60 Hz version will operate from 90–128 Vac, without modifications, and will use less than 4 A operating. The 50 Hz version 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 90–120 Vac and 184–240 Vac will use less than 4 A operating. The voltage ranges of 100–128 Vac and 200–256 Vac will use less than 2 A. Both versions of the RX02 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.

**2.1.3.2 Input Power Modification Requirements** – The 60 Hz version of the RX02 uses the H771-A power supply and will operate on 90–128 Vac, without modification. To convert to operate on a 50 Hz power source in the field, the H771-A supply must be replaced with an H771-C or -D (Figure 1-4) and the drive motor belt and drive motor pulley must be replaced (Figure 1-5). The H771-C operates on a 90–120 Vac or 100–128 Vac power source. The H771-D operates on a 184–240 Vac or 200-256 Vac power source. To convert the H771-C to the higher voltage ranges or the H771-D to the lower voltage ranges, the power harness and circuit breaker must be changed. See Figure 2-3 for the appropriate jumper and circuit breaker.



| VOLTAGE (VAC) | JUMPER | CIRCUIT BREAKER |
|---|---|---|
| 90-120 | 70-10696-02 | 3.5 A, 12-12301-01 |
| 100-128 | 70-10696-01 | 3.5 A, 12-123-1-01 |
| 184-240 | 70-10696-04 | 1.75 A, 12-12301-00 |
| 200-256 | 70-10696-03 | 1.75 A, 12-12301-00 |

MA-1855

Figure 2-3 RX02 Rear View

### 2.1.4 Fire and Safety Precautions

The RX02 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.

## 2.2 CONFIGURATION GUIDELINES

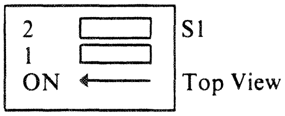
The most common RX02 Floppy Disk System configurations available are listed in Table 2-1. Each interface module listed in the table plugs into a computer bus; it is compatible with the applicable computer so that there is adequate power to operate each module. The interconnections between each interface module and the RX02 controller for each of the configurations in Table 2-1 is by a BC05L-15 cable which is 4.6 m (15 ft) maximum. (See Table 2-2 for the controller module configuration switch positions.)

Table 2-1 RX02 Configurations

| Computer | System Designation | μCPU Controller | Interface Module | RX02 Model No. | Power Supply |
|---|---|---|---|---|---|
| PDP-8 | RX8E | M7744 | M8357 | RX02-BA<br>RX02-BC<br>RX02-BD | 115 V, 60 Hz<br>115 V, 50 Hz<br>230 V, 50 Hz |
| | RX28E | M7744 | M8357 | RX02-BA<br>RX02-BC<br>RX02-BD | 115 V, 60 Hz<br>115 V, 50 Hz<br>230 V, 50 Hz |
| PDP-11 | RX11 | M7744 | M7846 | RX02-BA<br>RX02-BD<br>RX02-BD | 115 V, 60 Hz<br>230 V, 50 Hz<br>230 V, 50 Hz |
| | RX211 | M7744 | M8256 | RX02-BA<br>RX02-BC<br>RX02-BD | 115 V, 60 Hz<br>115 V, 50 Hz<br>230 V, 50 Hz |
| LSI-11 | RXV11 | M7744 | M7946 | RX02-BA<br>RX02-BC<br>RX02-BD | 115 V, 60 Hz<br>115 V, 50 Hz<br>230 V, 50 Hz |
| | RXV21 | M7744 | M8029 | RX02-BA<br>RX02-BC<br>RX02-BD | 115 V, 60 Hz<br>115 V, 50 Hz<br>230 V, 50 Hz |

Table 2-2 Controller Configuration Switch Positions

| Interface | S1-1 | S1-2 |
|---|---|---|
| RX211, RXV21, | OFF | ON |
| RX8E, RX11, RXV11, | ON | OFF |
| RX28 | OFF | OFF |



## 2.3 ENVIRONMENTAL CONSIDERATIONS

### 2.3.1 General

The RX02 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.

### 2.3.2 Temperature, Relative Humidity

The operating ambient temperature range of the diskette is 15° to 32° C (59° to 90° F) with a maximum temperature gradient of 11° C/hr (20° F/hr). The media nonoperating temperature range (storage) is increased to -34.4° to 51.6° C (-30° to 125° F), 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 51.6° C (125° F).

Humidity control is important in any system because static electricity can cause errors in any CPU with memory. The RX02 is designed to operate efficiently within a relative humidity range of 20 to 80 percent, with a maximum wet bulb temperature of 25° C (77° F) and a minimum dew point of 2° C (36° F).

### 2.3.3 Heat Dissipation

The heat dissipation factor for the RX02 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 RX02 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 RX02. 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 RX02 should not be subjected to unusually contaminated atmospheres, especially one with abrasive airborne particles.

NOTE

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

## 2.4 UNPACKING AND INSPECTION

### 2.4.1 General

The RX02 Floppy Disk System can be shipped in a cabinet as an integral part of a system or in a separate container. If the RX02 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 RX02 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

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

4. Remove the shipping restraint from the RX02 and save it for possible reuse.

5. Slide the RX02 out on the chassis slides and visually inspect for any damage as indicated in Paragraph 2.4.3.3.

#### 2.4.3.2 Separate Container

1. Open the carton (Figure 2-4) and remove the packing pieces.

2. Lift the RX02 out of the carton.

3. Remove the shipping fixtures from both sides of the RX02 and inspect for shipping damage as indicated in Paragraph 2.4.3.3.

4. Attach the inside tracks of the chassis slides provided in the carton to the RX02 (Figure 2-1).

5. Locate the proper holes in the cabinet rails (Figure 2-5) and attach the outside tracks to the cabinet.

6. Place the tracks attached to the RX02 inside the extended cabinet tracks and slide the unit in until the tracks lock in the extended position.

7. Attach the front bezel with the screws supplied.

8. Locate the RX02 cover in the cabinet above the unit and secure it to the cabinet rails (Figure 2-5).

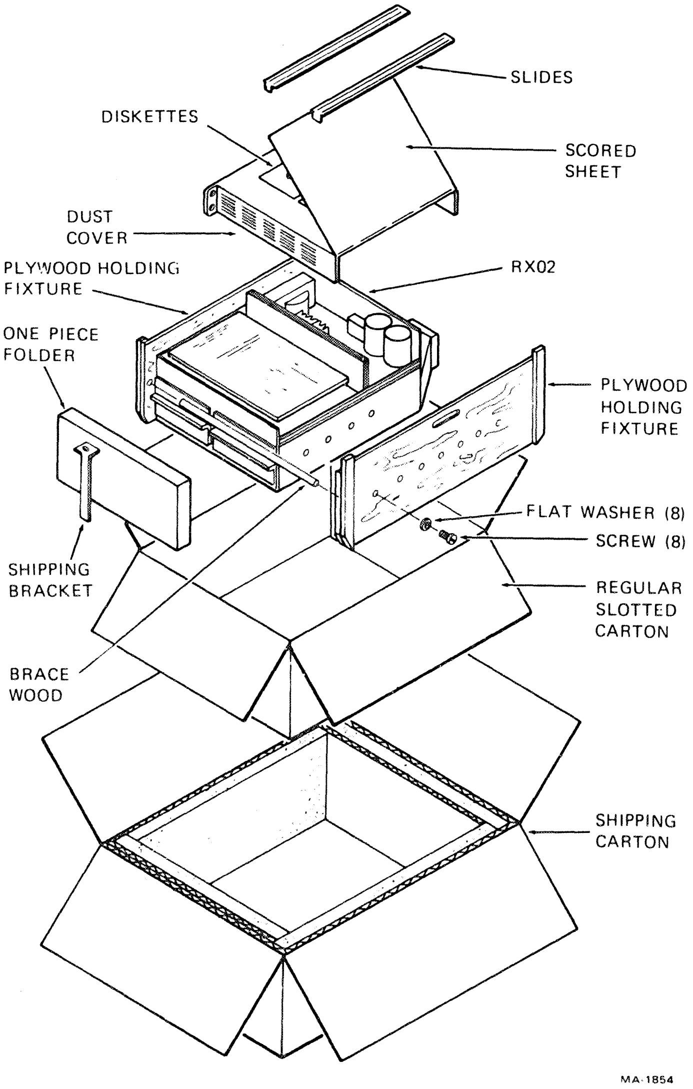


Figure 2-4 RX02 Unpacking

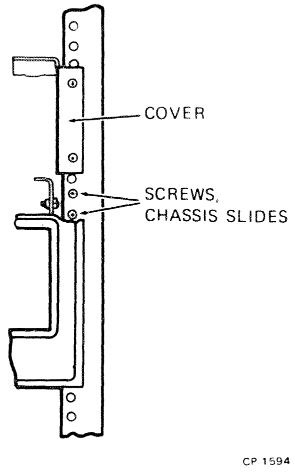


Figure 2-5 RX02 Cabinet Mounting Information

### 2.4.3.3 Inspection

1. Inspect the front cover(s) of the RX02 to be sure it operates freely. Compress the latch which allows the spring-loaded front cover to open.

2. Inspect the rear of the RX02 chassis to be sure there are no broken or bent plugs. Also, be sure the fuse is not damaged.

3. Visually inspect the interior of the unit for damaged wires or loose hardware.

4. Loosen the screws securing the hinged upper module (M7744) and raise the module so that modules M7744 and M7745 can be inspected for damaged components or wires.

5. Verify that the items listed on the shipping order are included in the shipment. Be sure the interface cable (BC05L-15) and the appropriate interface module are included.

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.**

## 2.5 INSTALLATION

1. Ensure that power for the system is off.

2. Loosen the screws securing the upper module (M7744) and swing it up on the hinge.

3. Inspect the wiring and connectors for proper routing and ensure that they are seated correctly.

4. This step is for 50 Hz versions only. Check the power configuration to ensure that the proper jumpers and the correct circuit breaker are installed (Figure 2-3).

5. Connect the BC05L-15 cable to the M7744 module and route it along the near side of the chassis through the back of the RX02 to the CPU; then connect it to the interface module for the PDP-8, PDP-11, or LSI-11.

   The cable is connected to the M7744 module with the red stripe on the left, looking from the component side of board; the cable is connected to the interface module with the red stripe toward the center of the module.

6. Refer to Table 2-2 for the correct controller configuration switch positions.

7. Refer to Table 2-3 for correct device code or addressing jumpers on the interface module.

8. Insert the interface module into the Omnibus (PDP-8), available SPC slot (PDP-11), or LSI bus (LSI-11). The PDP-11 and LSI-11 interface modules must be inserted in the lowest numbered available option location. Modules that use DMA processing should have a higher priority than programmed I/O devices. For modules using DMA processing in the PDP-11 SPC slot, ensure that the NPG (NPG IN, NPG OUT) line (CA1–CB1) is cut on the backplane.

9. Connect the RX02 ac power cord into a switched power source.

10. Turn the power on, watching for head movement on the drive(s) during the power up, initialize phase. The head(s) should move one track toward the center and back to track zero.

**Table 2-3 Interface Code/Jumper Configuration**

**PDP-8 (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 |

**PDP-11 (M7846) (M8256)**

| BR Priority | Unibus Address 17717X* | | Vector Address ($264_8$)* | |
|---|---|---|---|---|
| BR7 – 54-08782 | A12/W18 – Removed | SW10 OFF | V2/W1 – Installed | SW1 ON |
| BR6 – 54-08780 | A11/W17 – Removed | SW9 OFF | V3/W2 – Removed | SW2 OFF |
| BR5 – 54-08778* | A10/W16 – Removed | SW8 OFF | V4/W3 – Installed | SW3 ON |
| BR4 – 57-08776 | A9/W15 – Removed | SW7 OFF | V5/W4 – Installed | SW4 ON |
| | A8/W14 – Installed | SW6 ON | V6/W5 – Removed | SW5 OFF |
| | A7/W13 – Installed | SW5 ON | V7/W6 – Installed | SW6 ON |
| | A6/W12 – Removed | SW4 OFF | V8/W7 – Removed | SW7 OFF |
| | A5/W11 – Removed | SW3 OFF | | |
| | A4/W10 – Removed | SW2 OFF | | |
| | A3/W9 – Removed | SW1 OFF | | |

**LSI-11**

| (M7946) | | (M8029) | |
|---|---|---|---|
| **Register Address* ($17717X_8$)** | **Vector Address* ($264_8$)** | **Register Address* ($17717X_8$)** | **Vector Address* ($264_8$)** |
| A-1 – CPU Selectable | W-1/V-2 – Removed | A-1 – CPU Selectable | V2 – Installed |
| W-7/A-2 – Installed | W-2/V-3 – Installed | A-2 – Hardwired | V-3 – Removed |
| W-8/A-3 – Removed | W-3/V-4 – Removed | A-3 – Installed | V-4 – Installed |
| W-9/A-4 – Removed | W-4/V-5 – Removed | A-4 – Installed | V-5 – Installed |
| W-10/A-5 – Removed | W-5/V-6 – Installed | A-5 – Installed | V-6 – Removed |
| W-11/A-6 – Removed | W-6/V-7 – Removed | A-6 – Installed | V-7 – Installed |
| W-12/A-7 – Installed | | A-7 – Removed | V-8 – Removed |
| W-13/A-8 – Installed | | A-8 Removed | |
| W-14/A-9 – Removed | | A-9 – Installed | |
| W-15/A-10 – Removed | | A-10 – Installed | |
| W-16/A-11 – Removed | | A-11 – Installed | |
| W-17/A-12 – Removed | | A-12 – Installed | |

*Standard

### 2.5.1 PDP8-A Modification

In order to bootload from an RX02 on a PDP8-A system, it is necessary to modify the KM8-A (M8317) extended option module (if present) as follows (Figure 2-6):

- replace E82 with prom #23-465A2
- replace E87 with prom #23-469A2
- set SW#1 and SW#2 according to the bootload device as shown below.

| Program | S2-5 | S2-6 | S2-7 | S2-8 | S1-1 | S1-2 | S1-3 |
|---|---|---|---|---|---|---|---|
| H/L PTR | ON | ON | ON | OFF | ON | ON | ON |
| RK8-E | ON | OFF | ON | OFF | ON | OFF | ON |
| RX8-E | ON | OFF | OFF | ON | OFF | ON | ON |
| RL8A | OFF | ON | OFF | OFF | OFF | ON | OFF |

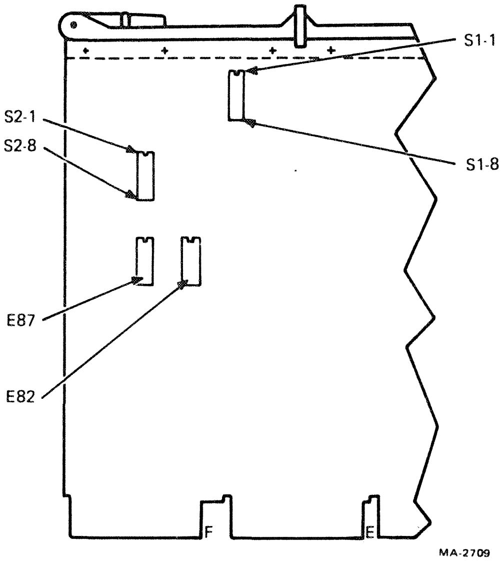


Figure 2-6 KM8-A Modification

## 2.6 TESTING

To test the operation of RX02, run the DEC diagnostics supplied. Perform the diagnostics in the sequence listed for the number of passes (time) indicated.

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

If any errors occur contact Field Service.

# CHAPTER 3
# USER INFORMATION

## 3.1 CUSTOMER RESPONSIBILITY

It is the user's responsibility to ensure that the RX02 is located and operated in an area that is free from excessive dust and dirt, and meets or exceeds the environmental conditions listed in Paragraph 1.4. The exterior of the RX02 should be kept clean. Also, it is the user's responsibility to ensure that the diskettes are handled and stored properly in order to prevent errors or data loss which might occur when recording or reading data; diskette handling procedures are described in Paragraph 3.2.

## 3.2 CARE OF MEDIA

### 3.2.1 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.

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 a 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 the operating range, allow five minutes for thermal stabilization before use. The diskette should be removed from its packaging during this time.

**CAUTION**

- **Do not use paper clips on diskettes.**
- **Do not expose the diskette to a heat source or sunlight.**
- **Keep the diskettes from magnetic fields.**
- **Do not write on the diskette with an instrument that leaves an impression or flakes.**

### 3.2.2 Diskette Storage

#### 3.2.2.1 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.

**3.2.2.2 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.

### 3.2.3 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 inches 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.

## 3.3 OPERATING INSTRUCTIONS

**NOTE**
**The left drive is always identified as drive 0.**

The RX02 has no operator controls and indicators. The diskette is inserted on a drive after compressing the latch to allow the spring-loaded front cover to open. Place the diskette with the label or top up (the jacket seams are on the bottom) on the drive spindle. Close the front cover which will automatically lock when it is pushed down. Initialize the system (from the computer) and listen for audible clicking sounds which indicate the head is moving over the diskette; the RX02 is ready for use. Data storage and retrieval is controlled by the user's program.

**CAUTION**
**Do not open the drive door while the diskette is in use; this results in errors.**

## 3.4 OPERATOR TROUBLESHOOTING

Table 3-1 is a list of possible problems and some probable causes the operator may encounter. If the problem cannot be corrected, refer to the *RX02 Floppy Disk System Technical Manual* if available.

**Table 3-1 Operator Troubleshooting Guide**

| Problem | Probable Cause | Correction |
|---|---|---|
| No power (drive inoperative) | a. Power cord disconnected<br>b. Blown fuse<br>c. Circuit breaker open | a. Connect power cord<br>b. Replace fuse<br>c. Close circuit breaker |
| Drive not ready | a. Drive door open<br>b. Diskette improperly installed | a. Close door<br>b. Properly seat diskette |
| Error in recording | a. Diskette wear<br>b. Diskette mounting hole<br>c. Mismatch in recording density on a diskette | a. If worn, replace<br>b. If the hole is not concentric, replace diskette<br>c. If diskette data density is not compatible with data to be recorded, replace diskette with a new preformatted diskette. |

# CHAPTER 4
# PROGRAMMING

This chapter contains programming information for the following interface options: RX8E, RX28, RX11, RXV11, RX211, and RXV21. The RX8E and RX28 programming information is presented followed by the RX11 and RXV11 information and then the RX211 and RXV21 information is presented. The RX8E, RX11, and RXV11 options are used for single density recording and are compatible with the RX01 Floppy Disk System. The RX28E, RX211, and RXV21 can be used for either single or double density recording.

## 4.1 RX8E AND RX28 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 2 sectors to store 1 page of information. The diskette capacity in this mode is 128,128 12-bit words (1001 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. The 8-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 ms after the previous request was serviced for 12-bit mode (18 ms for 8-bit mode). There is no maximum time between the transfer request from the RX02 and servicing of that request by the host processor. This allows the data transfer to and from the RX02 to be interrupted without loss of data.

The RX28 interface allows two modes of data transfer: 8-bit word length and 12-bit word length. For each mode of data transfer there can be either single density or double density storage of data. In the 12-bit mode single density recording, 64 words are written in a diskette sector, and the diskette capacity is 128,128 12-bit words; for double density, there are 128 words written in a sector with a diskette capacity of 256,256 12-bit words. In the 8-bit word mode single density recording, 128 8-bit bytes are written in each sector and the diskette capacity is 256,256 8-bit bytes; for double density, there are 256 8-bit bytes written in a sector with a diskette capacity of 512,512 8-bit bytes. (For the 12-bit mode, all 12-bit data words are loaded into the buffer and then the hardware forces zeros to add extra bits to the end of the buffer so that the buffer is filled.)

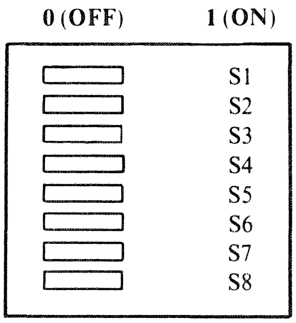
### 4.1.1 Device Codes

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/RX28 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/RX28 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/RX28 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 |



### 4.1.2 Instruction Set

The RX8E/RX28 instruction set is listed below and described in the following paragraphs. When operating as an RX28, for the 8-bit mode, all instruction set commands are transferred in two 8-bit bytes.

| 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 |

**4.1.2.1 RX8E Load Command (LCD) - 67x1** - This command transfers the contents of the AC to the interface register and clears the AC. The RX02 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 RX02 has completed the previous function. The command word is defined as shown in Figure 4-1. The command word is described in greater detail in Paragraph 4.1.3.1.

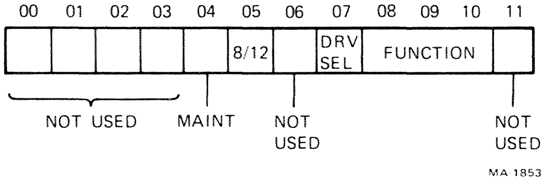


Figure 4-1 LCD Word Format (RX8E)

**4.1.2.2 RX28 Load Command - (First byte 67x1, second byte - 67x2)** - This command transfers the contents of the AC to the interface register and clears the AC. The RX02 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 RX02 has completed the previous function. The command word is defined as shown in Figure 4-2 and is described in greater detail in Paragraph 4.1.3.1.

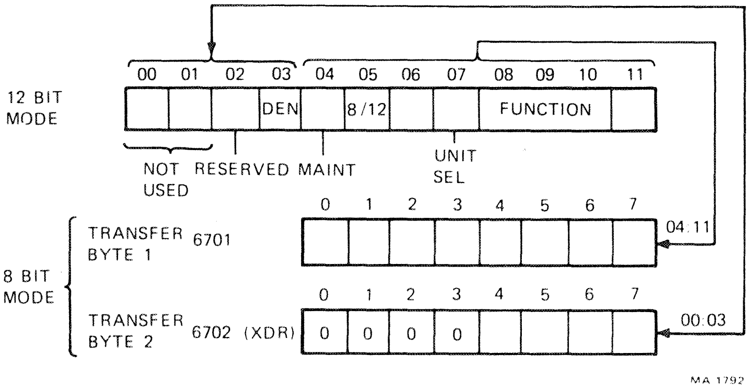


Figure 4-2 Command Word Format (RX28)

When operating in the 8-bit mode, the Load command is stored in two 8-bit transfers. The first 8 bits of the command word (shown as bits 4–11 in Figure 4-2) are stored; then TR is asserted and an XDR is performed to transfer the remaining bits of data (bit 3, DEN, and bit 2, as shown in Figure 4-2) right-justified. The extra bits in the second 8-bit transfer are filled with zeros. Upon completing the transfer of the second 8-bit byte, Done is asserted to end the function.

**4.1.2.3 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 transfer is governed by the RX02 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 RX02 that:

1. The last data word supplied by the RX02 has been accepted by the PDP-8, and the RX02 can proceed, or

2. The data or address word requested by the RX02 has been provided by the PDP-8, and the RX02 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 RX02. 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 RX02 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.

**4.1.2.4 STR - 67x3** - This instruction causes the next instruction to be skipped if the transfer request (TR) flag has been set by RX02 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.

**4.1.2.5 SER - 67x4** - This instruction causes the next instruction to be skipped if the error flag has been set by an error condition in the RX02 and clears the flag. An error also causes the done flag to be set (Paragraph 4.1.3.6).

**4.1.2.6 SDN - 67x5** - This instruction causes the next instruction to be skipped if the done flag has been set by the RX02, 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.

**4.1.2.7 INTR - 67x6** - This instruction enables interrupts by the done flag if AC 11=1. It disables interrupts if AC 11=0.

**4.1.2.8 INIT - 67x7** - The instruction initializes the RX02 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 RX02 returns to the Done state. Initialize can be generated by the program or by the Omnibus Initialize.

### 4.1.3 Register Description

Only one physical register (the interface register) exists in the RX8E/RX28, but it may represent one of the six RX02 registers described in the following paragraphs, according to the protocol of the function in progress.

**4.1.3.1 Command Register (Figures 4-3 and 4-4)** - The command is loaded into the interface register by the LCD instruction for RX8E and by a load command (LCD and XDR) for the RX28 (Paragraphs 4.1.2.1 and 4.1.2.2).

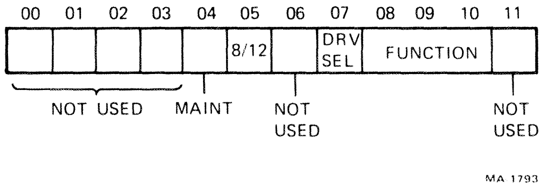


Figure 4-3 Command Register Format (RX8E)

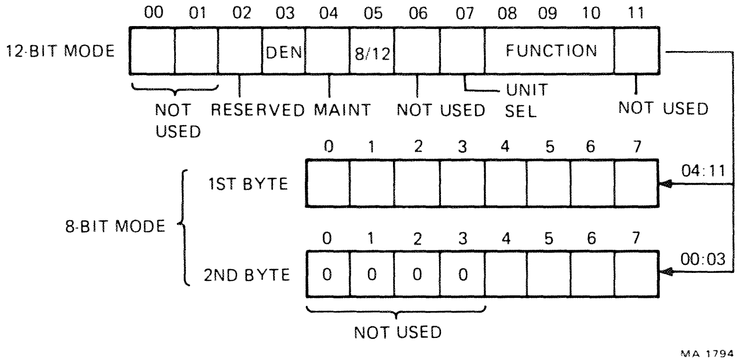


Figure 4-4 Command Register Format (RX28)

The function codes (bits 8, 9, 10) are summarized below and described in Paragraph 4.1.4.

| Code | Function |
|---|---|
| 000 | Fill Buffer |
| 001 | Empty Buffer |
| 010 | Write Sector |
| 011 | Read Sector |
| 100 | Not used (RX8E) - Set Density (RX28) |
| 101 | Read Status |
| 110 | Write Deleted Data Sector |
| 111 | Read Error Register |

The DRV (UNIT) 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 | 12-bit mode selected |
| AC 5 = 1 | 8-bit mode selected |

The DEN bit (bit 3) for RX28 indicates the density for the function to be performed (0 = single, 1 = double). The RX8E/RX28 will initialize into 12-bit mode.

**4.1.3.2 Error Code Register (Figure 4-5)** - Specific error codes can be accessed by use of the read error code function (111) (Paragraph 4.1.4.9). The specific octal error codes are given in Paragraph 4.1.5.

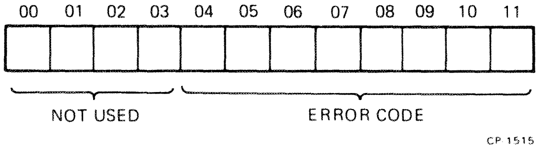


Figure 4-5 Error Code Register Format (RX8E/RX28A)

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 RX02; the on-line condition exists when the cable is connected to the RX02.

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 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 a Load Command 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 Load Command IOT, which sets the maintenance flip-flop. However, successive Load Command 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 on 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.

With the maintenance flip-flop set, it can be determined if the interface is capable of generating an interrupt on the Omnibus. When the maintenance flip-flop is set, the done flag is set, and 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. Exiting from the on-line maintenance bit mode should be finalized by an initialize to the RX02.

**4.1.3.3 RX2TA - RX Track Address (Figure 4-6)** - This register is loaded to indicate on which of the 77 (0–76) tracks a given function is to operate. It can be addressed only under the protocol of the function in progress (Paragraph 4.1.4). Bits 0–3 are unused and are ignored by the control.

**4.1.3.4 RX2SA - RX Sector Address (Figure 4-7)** - This register is loaded to indicate on which of the 26 (1–26) sectors a given function is to operate. It can be addressed only under the protocol of the function in progress (Paragraph 4.1.4). Bits 0–3 are unused and are ignored by the control.

**4.1.3.5 RX2DB - RX Data Buffer (Figure 4-8)** - 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 Load Command IOT is issued (Paragraph 4.1.3.1).

**4.1.3.6 RX8E - RX Error and Status (Figure 4-9)** - 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.

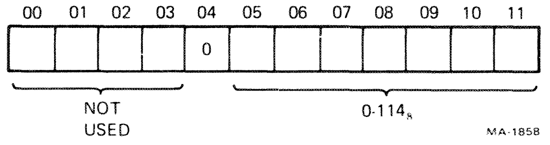


Figure 4-6 RX2TA Format (RX8E/RX28)

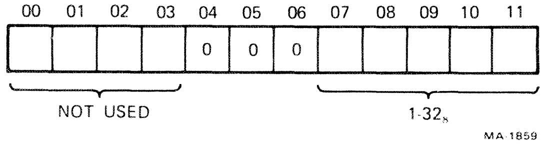


Figure 4-7 RX2SA Format (RX8E/RX28)

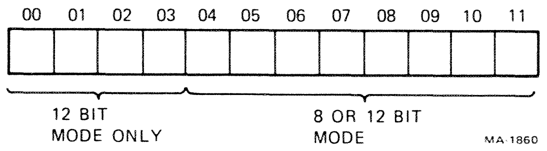


Figure 4-8 RX2DB Format (RX8E/RX28)

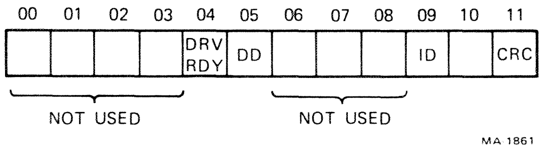


Figure 4-9 RXES Format (RX8E)

**Bit No.** **Description**

11 CRC Error - The cyclic redundancy check at the end of the data field has indicated an error. The data must be considered invalid; it is suggested that the data transfer be retried up to 10 times, as most data errors are recoverable (soft).

9 Initialize Done - This bit indicates completion of the Initialize routine. It can be asserted due to RX02 power failure, system power failure, or programmable or bus Initialize. This bit is not available within the RXES from a read status function.

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 RX02. 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 unsuccesful 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 RX2CS but error bits are not set in the RXES, specific error conditions can be accessed via a read error register function.**

**4.1.3.7 RX28 - RX Error and Status (Figure 4-10)** - The RX2ES 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 RX2ES is also available in the interface register upon completion of any function. The RX2ES is accessed by the XDR instruction. The meaning of the error bits is given below.

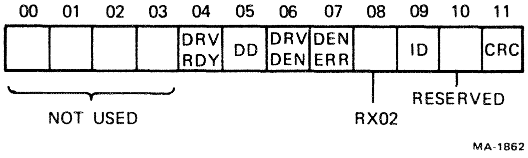


Figure 4-10 RX2ES Format (RX28)

**Bit No.** **Description**

11 CRC Error - The cyclic redundancy check at the end of the data field has indicated an error. The data must be considered invalid; it is suggested that the data transfer be retried up to 10 times; as most data errors are recoverable (soft).

10 Reserved.

9 Initialize Done - This bit indicates completion of the Initialize routine. It can be asserted due to RX02 power failure, system power failure, or programmable or bus Initialize. This bit is not available within the RX2ES from a read status function.

| Bit No. | Description |
|---|---|
| 8 | RX02 - This bit is asserted if an RX02 system is being used. |
| 7 | DEN ERR - This bit indicates that the density of the function does not agree with the drive density. Upon detection of this error the control terminates the operation and asserts error and done. |
| 6 | DRV DEN - This bit indicates the density of the diskette in the drive selected (0 = single, 1 = double). |
| 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 RX02. 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 RX2CS but error bits are not set in the RX2ES, specific error conditions can be accessed via a read error code function.**

### 4.1.4 Function Code Description

The RX8E/RX28 functions are initiated by means of the Load command described in Paragraphs 4.1.2.1 and 4.1.2.2. The done flag should be tested and cleared with the SDN instruction in order to verify that the RX8E/RX28 is in the Done state prior to issuing the command instruction. Upon receiving a command instruction while in the Done state, the RX8E/RX28 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. 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 (RX8E), Set Density (RX28) |
| 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.**

**4.1.4.1 Fill Buffer (000)** – For RX8E this function is used to load the RX02 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. For RX28 this function loads the sector buffer in 12-bit mode with 64 12-bit words for single density or 128 12-bit words for double density; in the 8-bit mode, the buffer is loaded with 128 8-bit bytes for single density or 256 8-bit bytes for double density. 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 RX02 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 4.1.2.4). The data word can then be transferred to the interface register by means of the XDR IOT. The RX02 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 RX2ES is moved to the interface register, and the RX02 sets the done flag to indicate the completion of the function. Therefore, it is unnecessary for the host processor to keep a count of the data transfers. Any XDR commands after Done is set will result in the RX2ES status word being loaded in the AC. The sector buffer must be completely loaded before the RX8E/RX28 will set Done and recognize a new command. An interrupt would now occur if Interrupt Enable were set.

**4.1.4.2 Empty Buffer (001)** – This function moves the contents of the sector buffer to the host processor. Upon decoding this function RX2ES bits 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/RX28 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 transfer for an XDR command is controlled by the RX02. The TR flag is set again with the next word in the interface register. The above sequence is repeated until all words or bytes have been transferred, thus emptying the sector buffer. The done flag is then set after the RX2ES 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.**

**4.1.4.3 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/RX28 clears the RX2ES 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 RX02 tests the supplied track address to determine if it is within the allowable limits. If it is not, the RX2ES 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 RX02 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 RX2ES to be loaded into the interface register, and the error and done flags to be set. If no errors are encountered, the RX2ES 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.**

**4.1.4.4 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/RX28 clears RX2ES 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 RX02. If illegal, the error and done flags are set with the RX2ES moved to the interface register and the function is terminated. Otherwise, the RX02 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 RX2ES 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 RX2ES in the interface register. A detection of CRC error results in RX2ES bit 11 being set. If a deleted data mark was encountered at the beginning of the desired data field, RX2ES bit 5 is set.

**4.1.4.5 Set Media Density (100) for RX28 only** - This function causes the entire diskette to be reassigned to a new density. The density bit (bit 3 RX2CS) indicates the new density of the diskette. The control reformats the diskette by writing new data address marks (double or single density) and zeroing out all data fields on the diskette. Before executing the command the control will look for a protective key word of 01001001 (ASCII'I').

The control starts at sector 1, track 0 and reads the header information, then starts a write operation, writing the new data address mark and data field as well as CRC characters. If the header information is damaged, the control will abort the operation and assert DONE and ERROR.

This operation takes about 15 seconds and should not be interrupted. If for any reason the operation is interrupted, an illegal diskette has been generated which may have data marks of both densities. This diskette should again be completely reformatted.

**4.1.4.6 Maintenance Read Status (101) for RX28 only** - This function updates the drive ready and drive density status of the selected drive, clears the INIT DONE bit, updates the Unit Sel, possibly sets the density error bit and leaves the remainder of the RX2ES unchanged. The drive density is updated by loading the head on the selected drive (without changing head and reading position) with the first header and data mark that randomly appears under the head. The control will then generate the appropriate number of shift pulses which will transfer the RX2ES (error and status) register over the interface. Upon completion of the RX2ES transfer, the control asserts Done to complete the operation.

**4.1.4.7 Read Status (101) for RX8E only** - Upon decoding this function, the RX02 moves the RXES to the RX8E interface register and sets the done flag. The RXES can then be read by the transfer data register (XDR) command. The bits are defined in Paragraph 4.1.3.6.

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

**4.1.4.8 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 1.5.3.2). RX2ES bit 5 (Deleted Data) will be set in the interface register upon completion of the function.

**4.1.4.9 Read Error Code Function (111)** – The read error code function can be used to retrieve explicit error information upon detection of the error flag. Upon receiving this function, the RX02 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 4.1.5).

NOTE

**Care should be exercised in the use of this function. The program must perform this function before a read status because the error register is always modified by a read status function.**

**4.1.4.10 Power Fail** – There is no actual function code associated with power fail. When the RX02 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 RX02 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 buffer.
4. Set Initialize Done bit of the RX2ES, 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.

INIT IOT is a method of aborting an incomplete function (Paragraph 4.1.2.7).

### 4.1.5 Error Recovery

**4.1.5.1 RX8E** – There are two error indications given by the RX8E system. The read status function (Paragraph 4.1.4.7) will assemble the current contents of the RXES (Paragraph 4.1.3.6), which can be sampled to determine errors. The read error register function (Paragraph 4.1.4.9) 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 READY 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 |
| 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 |
| 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 |
| 0220 | Self diagnostic error on Initialize |
| 0240 | Density Error |

**4.1.5.2 RX28** – There are two error indications given by the RX28 system. The read status function will assemble the current contents of the RX2ES which can be sampled to determine errors. The read error register function 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 RX2ES 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. |
| 0040 | Tried to access a track greater than 76 |
| 0050 | Home was found before desired track was reached. |
| 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 ID mark found within allowable time span |
| 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 |
| 0220 | R/W electronics failed maintenance mode test. |
| 0240 | Density error |
| 0250 | Wrong key word for Set Media Density command |

### 4.1.6 RX8E Programming Examples

**4.1.6.1 Write/Write Deleted Data/Read Functions** – Figure 4-11 presents a program for implementing a write, write deleted data, or a read function with interrupts turned off (IOF). The first 3 steps preset the PTRY, CTRY, and STRY retry counters, which are set at 10 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, the error is a seek error; 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 4.1.3.6). 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 10 tries, it is considered a hard error. If 10 retries have not occurred, a branch is made to RETRY and the sequence is repeated.

After a parity test, 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 10 retries, it is considered a hard error. If 10 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 4.1.2.8). 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 4-11, but is presented in Figure 4-13. After the system has been initialized, the program increments and tests the STRY retry counter. If a seek error persists after 10 tries, it is considered a hard error. If 10 retries have not occurred, a branch is made to RETRY and the sequence repeated.

**4.1.6.2 Empty Buffer Function** – Figure 4-12 shows a program for implementing an empty buffer function with interrupts turned off (IOF). The first instruction sets the number of retries at 10. 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 10 times. If the error persists, a hard parity error is assumed, indicating a problem in the interface cable.

**4.1.6.3 Fill Buffer Function** – Figure 4-13 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 INITIZLIAE 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
```

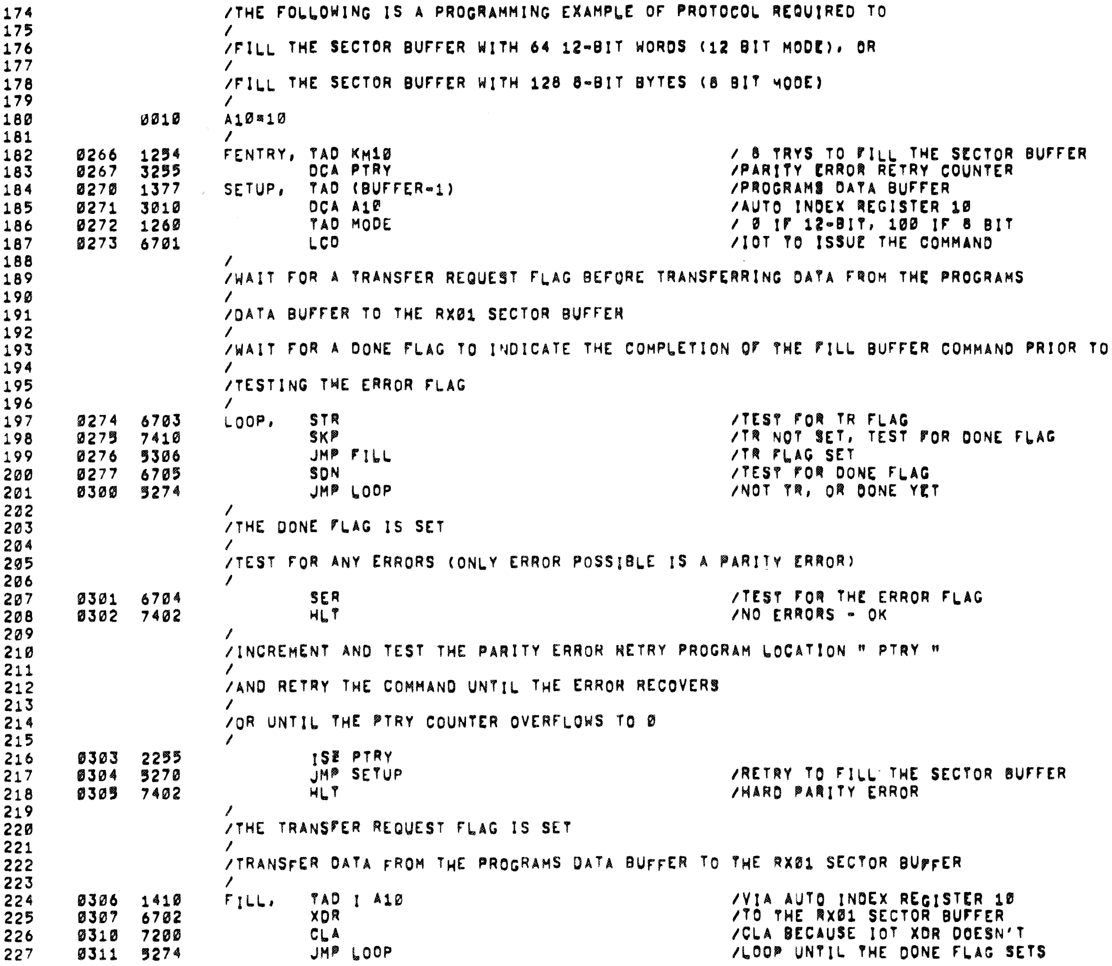
Figure 4-11 RX8E Write/Write Deleted Data/Read Example (Sheet 1 of 2)

```
                    /A CRC ERROR HAS OCCURED
                    /
                    /INCREMENT AND TEST THE CRC ERROR RETHY COUNTER PROGRAM LOCATION " CTRY "
                    /
                    /AND RETRY THE COMMAND UNTIL THE CRC ERROR RECOVERS
                    /
                    /OR UNTIL THE CTRY COUNTER OVERFLOWS TO Ø
                    /
Ø245  2256                  ISZ CTRY
Ø246  52Ø6                  JMP RETRY                            /RETRY THE COMMAND
Ø247  74Ø2                  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 1Ø, AND 11 = Ø)
                    /
Ø25Ø  67Ø7          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 Ø
                    /
Ø251  2257                  ISZ STRY
Ø252  52Ø6                  JMP RETRY                            /RETRY THE COMMAND
Ø253  74Ø2                  HLT                                  /HARD SEEK ERROR

                    /THE FOLLOWING PROGRAM LOCATIONS ARE REFERENCED WITHIN THIS EXAMPLE
                    /
Ø254  777Ø          KM1Ø,   -1Ø
                    /
                    /THE FOLLOWING 3 PROGRAM LOCATIONS ARE THE ERROR RETRY COUNTERS
                    /
Ø255  ØØØØ          PTRY,   Ø                                    /PARITY ERROR RETRY COUNTER
Ø256  ØØØØ          CTRY,   Ø                                    /CRC ERROR RETRY COUNTER
Ø257  ØØØØ          STRY,   Ø                                    /SEEK ERROR RETRY COUNTER
                    /
                    /PROGRAM LOCATION " MODE " CONTAINS A Ø IF 12-BIT MODE, OR
                    /CONTAINS A 1ØØ IF 8-BIT MODE
                    /
Ø26Ø  ØØØØ          MODE,   Ø                                    / Ø OR 1ØØ
                    /
                    /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)
                    /
Ø261  ØØØØ          COMMAND, Ø                                   / 4, 14, OR 6, OR 2
                    /
                    /PROGRAM LOCATION " UNIT " CONTAINS THE UNIT DESIGNATION
                    /
                    /UNIT Ø (Ø), OR UNIT 1 (2Ø)
                    /
Ø262  ØØØØ          UNIT,   Ø                                    / Ø, OR 2Ø
                    /
                    /PROGRAM LOCATION " SECTOR " CONTAINS THE SECTOR ADDRESS (1 TO 32 OCTAL)
                    /
Ø263  Ø2ØØ          SECTOR, Ø                                    / 1 TO 32 OCTAL
                    /
                    /PROGRAM LOCATION " TRACK " CONTAINS THE TRACK ADDRESS (Ø TO 114 OCTAL)
                    /
Ø264  ØØØØ          TRACK,  Ø                                    / Ø 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
                    /
                    / = Ø IF SEEK ERROR, 1 IF CRC ERROR, 2 IF PARITY ERROR
                    /
Ø265  ØØØØ          ASTATUS, Ø                                   /STATUS AT ERROR
```

Figure 4-11   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)
                 /
Ø312  1254       EENTRY, TAD KM1Ø                       / 8 TRYS TO EMPTY THE SECTOR BUFFER
Ø313  3255               DCA PTRY                       /PARITY ERROR RETRY COUNTER
Ø314  1377       ESETUP, TAD (BUFFER-1)                 /PROGRAMS DATA BUFFER
Ø315  3Ø1Ø               DCA A1Ø                        /AUTO INDEX REGISTER 1Ø
Ø316  126Ø               TAD MODE                       / Ø IF 12-BIT, 1ØØ IF 8 BIT
Ø317  1261               TAD COMMAND                    / 2 MEANS EMPTY BUFFER
Ø32Ø  67Ø1               LCD                            /IOT TO ISSUE THE COMMAND
                 /
                 /WAIT FOR A TRANSFER REQUEST FLAG BEFORE TRANSFERRING DATA TO THE PROGRAMS
                 /
                 /DATA BUFFER FROM THE RXØ1 SECTOR BUFFER
                 /
                 /WAIT FOR A DONE FLAG TO INDICATE THE COMPLETION OF THE EMPTY BUFFER COMMAND PRIOR TO
                 /
                 /TESTING THE ERROR FLAG
                 /
Ø321  67Ø3       ELOOP,  STR                            /TEST FOR TR FLAG
Ø322  741Ø               SKP                            /TR NOT SET, TEST FOR DONE FLAG
Ø323  5333               JMP EMPTY                      /TR FLAG SET
Ø324  67Ø5               SDN                            /TEST FOR DONE FLAG
Ø325  5274               JMP ELOOP                      /NOT TR, OR DONE YET
                 /
                 /THE DONE FLAG IS SET
                 /
                 /TEST FOR ANY ERRORS (ONLY ERROR POSSIBLE IS A PARITY ERROR)
                 /
Ø326  67Ø4               SER                            /TEST FOR THE ERROR FLAG
Ø327  74Ø2               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 Ø
                 /
Ø33Ø  2255               ISZ PTRY
Ø331  5314               JMP ESETUP                     /RETRY TO EMPTY THE SECTOR BUFFER
Ø332  74Ø2               HLT                            /HARD PARITY ERROR
                 /
                 /THE TRANSFER REQUEST FLAG IS SET
                 /
                 /TRANSFER DATA TO THE PROGRAMS DATA BUFFER FROM THE RXØ1 SECTOR BUFFER
                 /
Ø333  67Ø2       EMPTY,  XDR                            /FROM THE RXØ1 SECTOR BUFFER
Ø334  341Ø               DCA I A1Ø                      /TO THE PROGRAMS DATA BUFFER
Ø335  5321               JMP ELOOP                      /LOOP UNTIL THE DONE FLAG SETS
Ø377  Ø377
      Ø4ØØ               PAGE
                 /THE FOLLOWING PROGRAM LOCATIONS ARE RESERVED FOR THE PROGRAMS DATA BUFFER
                 /
Ø4ØØ  ØØØØ       BUFFER, Ø
      Ø6ØØ               *BUFFER+2ØØ
                         $
```

Figure 4-12  RX8E Empty Buffer Example

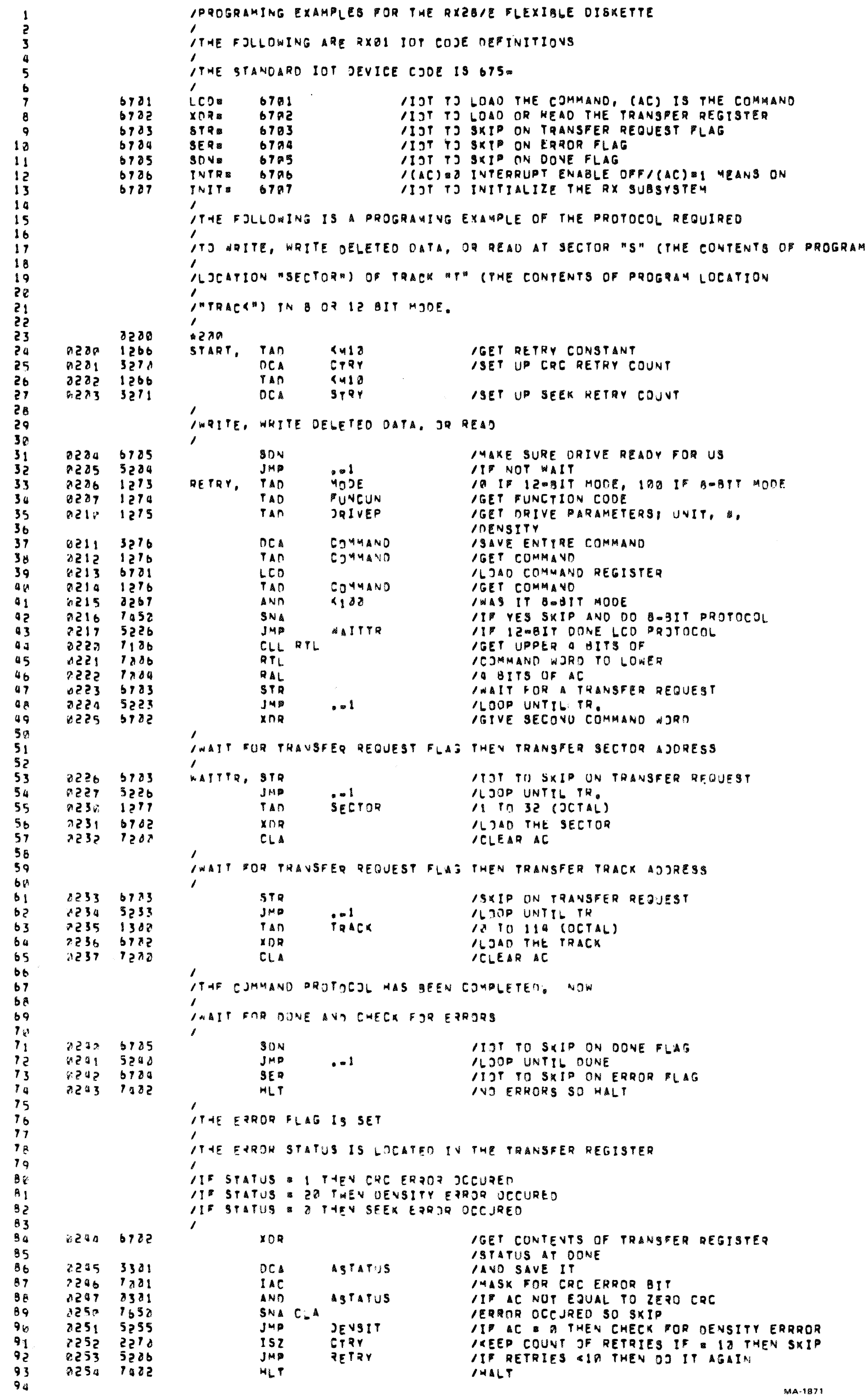


```
                   /THE FOLLOWING IS A PROGRAMMING EXAMPLE OF PROTOCOL REQUIRED TO
                   /
                   /FILL THE SECTOR BUFFER WITH 64 12-BIT WORDS (12 BIT MODE), OR
                   /
                   /FILL THE SECTOR BUFFER WITH 128 8-BIT BYTES (8 BIT MODE)
                   /
      ØØ1Ø         A1Ø=1Ø
                   /
Ø266  1254         FENTRY, TAD KM1Ø                        / 8 TRYS TO FILL THE SECTOR BUFFER
Ø267  3255                 DCA PTRY                        /PARITY ERROR RETRY COUNTER
Ø27Ø  1377         SETUP,  TAD (BUFFER-1)                  /PROGRAMS DATA BUFFER
Ø271  3Ø1Ø                 DCA A1Ø                         /AUTO INDEX REGISTER 1Ø
Ø272  126Ø                 TAD MODE                        / Ø IF 12-BIT, 1ØØ IF 8 BIT
Ø273  67Ø1                 LCD                             /IOT TO ISSUE THE COMMAND
                   /
                   /WAIT FOR A TRANSFER REQUEST FLAG BEFORE TRANSFERRING DATA FROM THE PROGRAMS
                   /
                   /DATA BUFFER TO THE RXØ1 SECTOR BUFFER
                   /
                   /WAIT FOR A DONE FLAG TO INDICATE THE COMPLETION OF THE FILL BUFFER COMMAND PRIOR TO
                   /
                   /TESTING THE ERROR FLAG
                   /
Ø274  67Ø3         LOOP,   STR                             /TEST FOR TR FLAG
Ø275  741Ø                 SKP                             /TR NOT SET, TEST FOR DONE FLAG
Ø276  53Ø6                 JMP FILL                        /TR FLAG SET
Ø277  67Ø5                 SDN                             /TEST FOR DONE FLAG
Ø3ØØ  5274                 JMP LOOP                        /NOT TR, OR DONE YET
                   /
                   /THE DONE FLAG IS SET
                   /
                   /TEST FOR ANY ERRORS (ONLY ERROR POSSIBLE IS A PARITY ERROR)
                   /
Ø3Ø1  67Ø4                 SER                             /TEST FOR THE ERROR FLAG
Ø3Ø2  74Ø2                 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 Ø
                   /
Ø3Ø3  2255                 ISZ PTRY
Ø3Ø4  527Ø                 JMP SETUP                       /RETRY TO FILL THE SECTOR BUFFER
Ø3Ø5  74Ø2                 HLT                             /HARD PARITY ERROR
                   /
                   /THE TRANSFER REQUEST FLAG IS SET
                   /
                   /TRANSFER DATA FROM THE PROGRAMS DATA BUFFER TO THE RXØ1 SECTOR BUFFER
                   /
Ø3Ø6  141Ø         FILL,   TAD I A1Ø                       /VIA AUTO INDEX REGISTER 1Ø
Ø3Ø7  67Ø2                 XDR                             /TO THE RXØ1 SECTOR BUFFER
Ø31Ø  72ØØ                 CLA                             /CLA BECAUSE IOT XDR DOESN'T
Ø311  5274                 JMP LOOP                        /LOOP UNTIL THE DONE FLAG SETS
```

Figure 4-13 RX8E Fill Buffer Example

### 4.1.7 RX28 Programming Examples

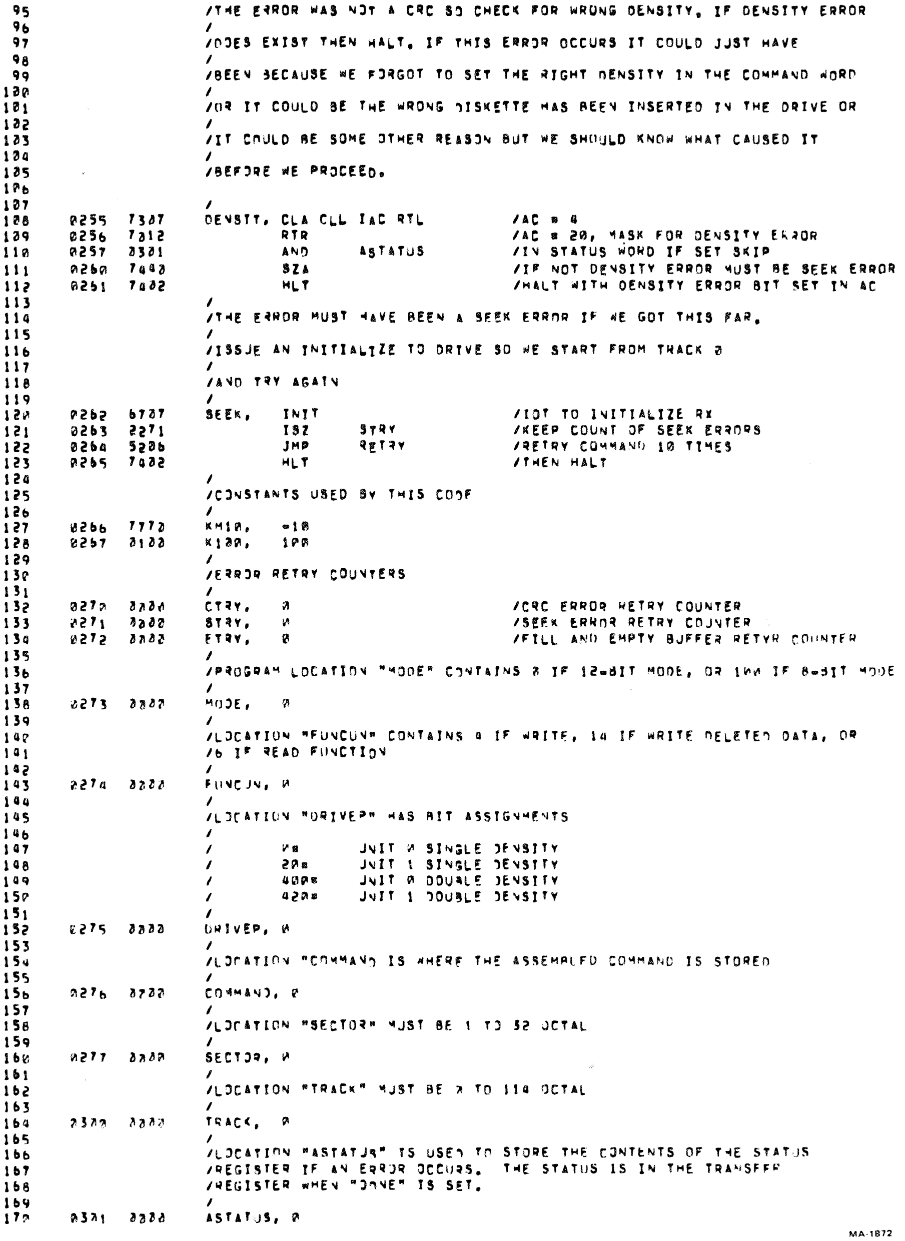
Figures 4-14, 4-15, and 4-16 are programming examples for write, write deleted data or read functions, for fill buffer functions, and for empty buffer functions, respectively. These examples are very similar to the RX8E programming examples described in Paragraph 4.1.6. Basically, there are two differences between the RX8E and RX28 examples. First, for the RX28 when a command is transferred in the 8-bit mode of operation, it is transferred in two 8-bit words using an XDR to transfer the second command word (see location 0225 in Figure 4-14); second, for the RX28, there is no parity error check as there is in the RX8E; instead there is a density error check.



```
                   /PROGRAMING EXAMPLES FOR THE RX28/E FLEXIBLE DISKETTE
                   /
                   /THE FOLLOWING ARE RX01 IOT CODE DEFINITIONS
                   /
                   /THE STANDARD IOT DEVICE CODE IS 675-
                   /
            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 TRANSFER REQUEST FLAG
            6704   SER=     6704             /IOT TO SKIP ON ERROR FLAG
            6705   SDN=     6705             /IOT TO SKIP ON DONE FLAG
            6706   INTR=    6706             /(AC)=0 INTERRUPT ENABLE OFF/(AC)=1 MEANS ON
            6707   INIT=    6707             /IOT TO INITIALIZE THE RX SUBSYSTEM
                   /
                   /THE FOLLOWING IS A PROGRAMING 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   *200
     0200   1266   START,   TAD     K10            /GET RETRY CONSTANT
     0201   3270            DCA     CTRY           /SET UP CRC RETRY COUNT
     0202   1266            TAD     K10
     0203   3271            DCA     STRY           /SET UP SEEK RETRY COUNT
                   /
                   /WRITE, WRITE DELETED DATA, OR READ
                   /
     0204   6705            SDN                    /MAKE SURE DRIVE READY FOR US
     0205   5204            JMP     .-1            /IF NOT WAIT
     0206   1273   RETRY,   TAD     MODE           /0 IF 12-BIT MODE, 100 IF 8-BIT MODE
     0207   1274            TAD     FUNCUN         /GET FUNCTION CODE
     0210   1275            TAD     DRIVEP         /GET DRIVE PARAMETERS; UNIT, #,
                                                   /DENSITY
     0211   3276            DCA     COMMAND        /SAVE ENTIRE COMMAND
     0212   1276            TAD     COMMAND        /GET COMMAND
     0213   6701            LCD                    /LOAD COMMAND REGISTER
     0214   1276            TAD     COMMAND        /GET COMMAND
     0215   0267            AND     K100           /WAS IT 8-BIT MODE
     0216   7452            SNA                    /IF YES SKIP AND DO 8-BIT PROTOCOL
     0217   5226            JMP     WAITTR         /IF 12-BIT DONE LCD PROTOCOL
     0220   7106            CLL RTL                /GET UPPER 4 BITS OF
     0221   7006            RTL                    /COMMAND WORD TO LOWER
     0222   7004            RAL                    /4 BITS OF AC
     0223   6703            STR                    /WAIT FOR A TRANSFER REQUEST
     0224   5223            JMP     .-1            /LOOP UNTIL TR.
     0225   6702            XDR                    /GIVE SECOND COMMAND WORD
                   /
                   /WAIT FOR TRANSFER REQUEST FLAG THEN TRANSFER SECTOR ADDRESS
                   /
     0226   6703   WAITTR,  STR                    /IOT TO SKIP ON TRANSFER REQUEST
     0227   5226            JMP     .-1            /LOOP UNTIL TR.
     0230   1277            TAD     SECTOR         /1 TO 32 (OCTAL)
     0231   6702            XDR                    /LOAD THE SECTOR
     0232   7200            CLA                    /CLEAR AC
                   /
                   /WAIT FOR TRANSFER REQUEST FLAG THEN TRANSFER TRACK ADDRESS
                   /
     0233   6703            STR                    /SKIP ON TRANSFER REQUEST
     0234   5233            JMP     .-1            /LOOP UNTIL TR
     0235   1300            TAD     TRACK          /0 TO 114 (OCTAL)
     0236   6702            XDR                    /LOAD THE TRACK
     0237   7200            CLA                    /CLEAR AC
                   /
                   /THE COMMAND PROTOCOL HAS BEEN COMPLETED.  NOW
                   /
                   /WAIT FOR DONE AND CHECK FOR ERRORS
                   /
     0240   6705            SDN                    /IOT TO SKIP ON DONE FLAG
     0241   5240            JMP     .-1            /LOOP UNTIL DONE
     0242   6704            SER                    /IOT TO SKIP ON ERROR FLAG
     0243   7402            HLT                    /NO ERRORS SO HALT
                   /
                   /THE ERROR FLAG IS SET
                   /
                   /THE ERROR STATUS IS LOCATED IN THE TRANSFER REGISTER
                   /
                   /IF STATUS = 1 THEN CRC ERROR OCCURED
                   /IF STATUS = 20 THEN DENSITY ERROR OCCURED
                   /IF STATUS = 0 THEN SEEK ERROR OCCURED
                   /
     0244   6702            XDR                    /GET CONTENTS OF TRANSFER REGISTER
                                                   /STATUS AT DONE
     0245   3301            DCA     ASTATUS        /AND SAVE IT
     0246   7001            IAC                    /MASK FOR CRC ERROR BIT
     0247   0301            AND     ASTATUS        /IF AC NOT EQUAL TO ZERO CRC
     0250   7650            SNA CLA                /ERROR OCCURED SO SKIP
     0251   5255            JMP     DENSIT         /IF AC = 0 THEN CHECK FOR DENSITY ERRROR
     0252   2270            ISZ     CTRY           /KEEP COUNT OF RETRIES IF = 10 THEN SKIP
     0253   5206            JMP     RETRY          /IF RETRIES <10 THEN DO IT AGAIN
     0254   7402            HLT                    /HALT
```

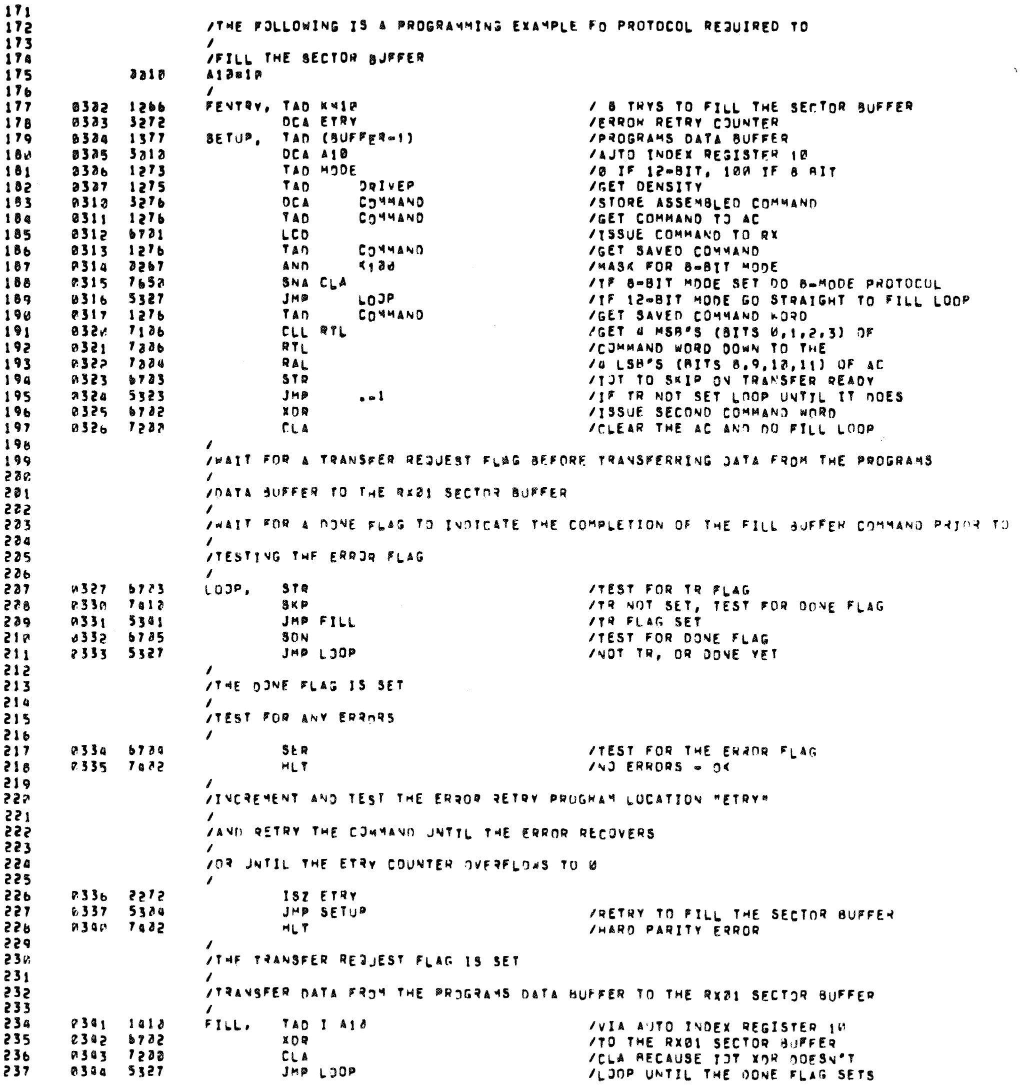
MA-1871

Figure 4-14 RX28 Write/Write Deleted Data/Read Example (Sheet 2 of 2)



```
                /THE ERROR WAS NOT A CRC SO CHECK FOR WRONG DENSITY, IF DENSITY ERROR
                /
                /DOES EXIST THEN HALT, IF THIS ERROR OCCURS IT COULD JUST HAVE
                /
                /BEEN BECAUSE WE FORGOT TO SET THE RIGHT DENSITY IN THE COMMAND WORD
                /
                /OR IT COULD BE THE WRONG DISKETTE HAS BEEN INSERTED IN THE DRIVE OR
                /
                /IT COULD BE SOME OTHER REASON BUT WE SHOULD KNOW WHAT CAUSED IT
                /
                /BEFORE WE PROCEED.

                /
0255  7307      DENSIT, CLA CLL IAC RTL          /AC = 4
0256  7012              RTR                      /AC = 20, MASK FOR DENSITY ERROR
0257  0301              AND     ASTATUS          /IN STATUS WORD IF SET SKIP
0260  7440              SZA                      /IF NOT DENSITY ERROR MUST BE SEEK ERROR
0261  7402              HLT                      /HALT WITH DENSITY ERROR BIT SET IN AC
                /
                /THE ERROR MUST HAVE BEEN A SEEK ERROR IF WE GOT THIS FAR.
                /
                /ISSUE AN INITIALIZE TO DRIVE SO WE START FROM TRACK 0
                /
                /AND TRY AGAIN
                /
0262  6707      SEEK,   INIT                     /IOT TO INITIALIZE RX
0263  2271              ISZ     STRY             /KEEP COUNT OF SEEK ERRORS
0264  5206              JMP     RETRY            /RETRY COMMAND 10 TIMES
0265  7402              HLT                      /THEN HALT
                /
                /CONSTANTS USED BY THIS CODE
                /
0266  7770      KM10,   -10
0267  0100      K100,   100
                /
                /ERROR RETRY COUNTERS
                /
0270  0000      CTRY,   0                        /CRC ERROR RETRY COUNTER
0271  0000      STRY,   0                        /SEEK ERROR RETRY COUNTER
0272  0000      ETRY,   0                        /FILL AND EMPTY BUFFER RETYR COUNTER
                /
                /PROGRAM LOCATION "MODE" CONTAINS 0 IF 12-BIT MODE, OR 100 IF 8-BIT MODE
                /
0273  0000      MODE,   0
                /
                /LOCATION "FUNCUN" CONTAINS 4 IF WRITE, 14 IF WRITE DELETED DATA, OR
                /6 IF READ FUNCTION
                /
0274  0000      FUNCUN, 0
                /
                /LOCATION "DRIVEP" HAS BIT ASSIGNMENTS
                /
                /       0=      UNIT 0 SINGLE DENSITY
                /       20=     UNIT 1 SINGLE DENSITY
                /       400=    UNIT 0 DOUBLE DENSITY
                /       420=    UNIT 1 DOUBLE DENSITY
                /
0275  0000      DRIVEP, 0
                /
                /LOCATION "COMMAND IS WHERE THE ASSEMBLED COMMAND IS STORED
                /
0276  0000      COMMAND, 0
                /
                /LOCATION "SECTOR" MUST BE 1 TO 32 OCTAL
                /
0277  0000      SECTOR, 0
                /
                /LOCATION "TRACK" MUST BE 0 TO 114 OCTAL
                /
0300  0000      TRACK,  0
                /
                /LOCATION "ASTATUS" IS USED TO STORE THE CONTENTS OF THE STATUS
                /REGISTER IF AN ERROR OCCURS.  THE STATUS IS IN THE TRANSFER
                /REGISTER WHEN "DONE" IS SET.
                /
0301  0000      ASTATUS, 0
```

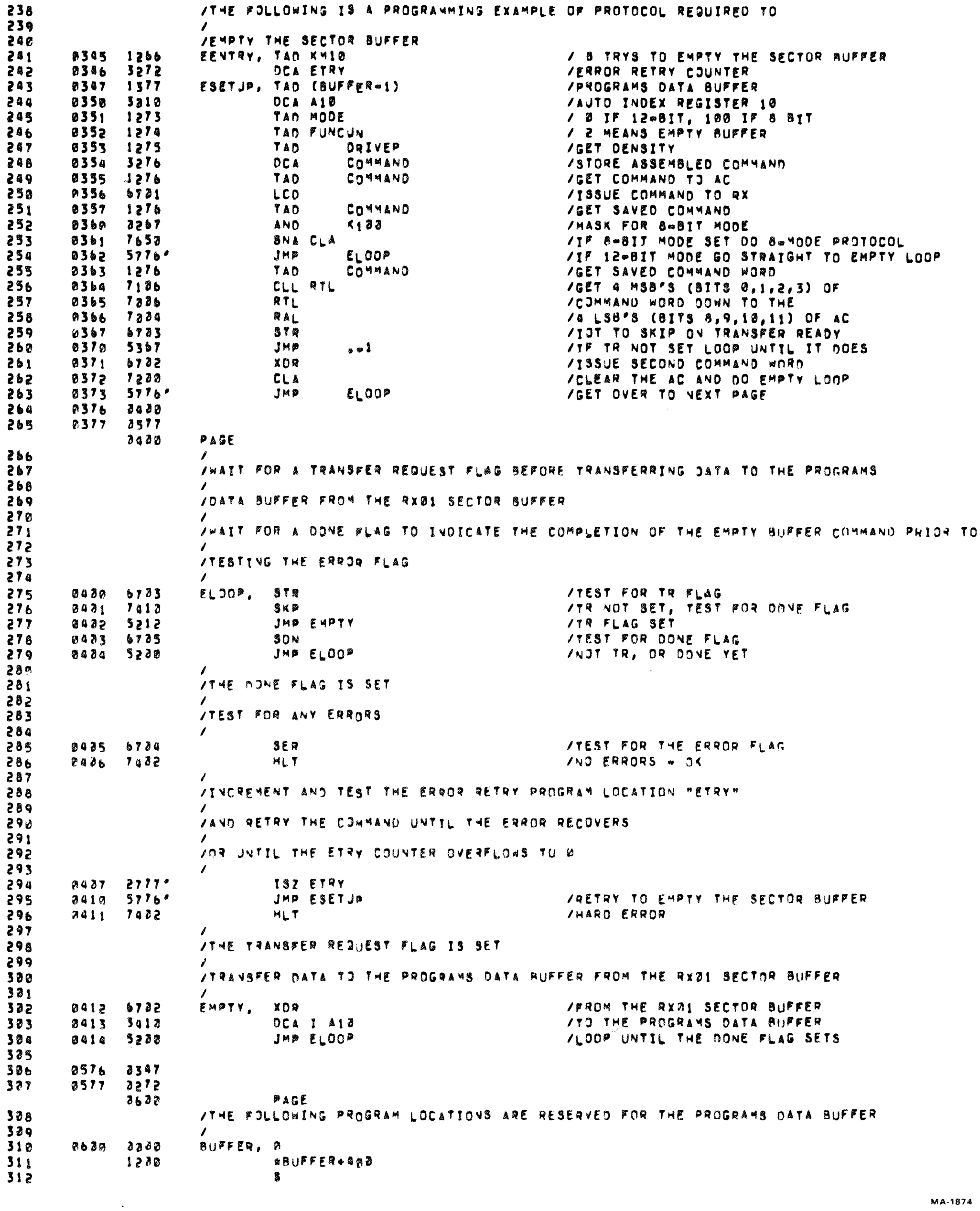
Figure 4-14 RX28 Write/Write Deleted Data/Read Example
(Sheet 2 of 2)



```

                      /THE FOLLOWING IS A PROGRAMMING EXAMPLE FO PROTOCOL REQUIRED TO
                      /
                      /FILL THE SECTOR BUFFER
            0010      A10=10
                      /
      0302  1266      FENTRY, TAD KM10                           / 8 TRYS TO FILL THE SECTOR BUFFER
      0303  3272              DCA ETRY                           /ERROR RETRY COUNTER
      0304  1377      SETUP,  TAD (BUFFER-1)                     /PROGRAMS DATA BUFFER
      0305  3010              DCA A10                            /AUTO INDEX REGISTER 10
      0306  1273              TAD MODE                           /0 IF 12-BIT, 100 IF 8 BIT
      0307  1275              TAD     DRIVEP                     /GET DENSITY
      0310  3276              DCA     COMMAND                    /STORE ASSEMBLED COMMAND
      0311  1276              TAD     COMMAND                    /GET COMMAND TO AC
      0312  6701              LCD                                /ISSUE COMMAND TO RX
      0313  1276              TAD     COMMAND                    /GET SAVED COMMAND
      0314  0267              AND     K100                       /MASK FOR 8-BIT MODE
      0315  7650              SNA CLA                            /IF 8-BIT MODE SET DO 8-MODE PROTOCOL
      0316  5327              JMP     LOOP                       /IF 12-BIT MODE GO STRAIGHT TO FILL LOOP
      0317  1276              TAD     COMMAND                    /GET SAVED COMMAND WORD
      0320  7106              CLL RTL                            /GET 4 MSB'S (BITS 0,1,2,3) OF
      0321  7006              RTL                                /COMMAND WORD DOWN TO THE
      0322  7004              RAL                                /4 LSB'S (BITS 8,9,10,11) OF AC
      0323  6703              STR                                /IOT TO SKIP ON TRANSFER READY
      0324  5323              JMP     .-1                        /IF TR NOT SET LOOP UNTIL IT DOES
      0325  6702              XDR                                /ISSUE SECOND COMMAND WORD
      0326  7200              CLA                                /CLEAR THE AC AND DO FILL LOOP
                      /
                      /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
                      /
      0327  6703      LOOP,   STR                                /TEST FOR TR FLAG
      0330  7410              SKP                                /TR NOT SET, TEST FOR DONE FLAG
      0331  5341              JMP FILL                           /TR FLAG SET
      0332  6705              SDN                                /TEST FOR DONE FLAG
      0333  5327              JMP LOOP                           /NOT TR, OR DONE YET
                      /
                      /THE DONE FLAG IS SET
                      /
                      /TEST FOR ANY ERRORS
                      /
      0334  6704              SER                                /TEST FOR THE ERROR FLAG
      0335  7402              HLT                                /NO ERRORS - OK
                      /
                      /INCREMENT AND TEST THE ERROR RETRY PROGRAM LOCATION "ETRY"
                      /
                      /AND RETRY THE COMMAND UNTIL THE ERROR RECOVERS
                      /
                      /OR UNTIL THE ETRY COUNTER OVERFLOWS TO 0
                      /
      0336  2272              ISZ ETRY
      0337  5304              JMP SETUP                          /RETRY TO FILL THE SECTOR BUFFER
      0340  7402              HLT                                /HARD PARITY ERROR
                      /
                      /THE TRANSFER REQUEST FLAG IS SET
                      /
                      /TRANSFER DATA FROM THE PROGRAMS DATA BUFFER TO THE RX01 SECTOR BUFFER
                      /
      0341  1410      FILL,   TAD I A10                          /VIA AUTO INDEX REGISTER 10
      0342  6702              XDR                                /TO THE RX01 SECTOR BUFFER
      0343  7200              CLA                                /CLA BECAUSE IOT XDR DOESN'T
      0344  5327              JMP LOOP                           /LOOP UNTIL THE DONE FLAG SETS
```

MA-1873

Figure 4-15 RX28 Fill Buffer Example



```
                    /THE FOLLOWING IS A PROGRAMMING EXAMPLE OF PROTOCOL REQUIRED TO
                    /
                    /EMPTY THE SECTOR BUFFER
0345  1266          EENTRY, TAD KM10                      / 8 TRYS TO EMPTY THE SECTOR BUFFER
0346  3272                  DCA ETRY                      /ERROR RETRY COUNTER
0347  1377          ESETUP, TAD (BUFFER-1)                /PROGRAMS DATA BUFFER
0350  3010                  DCA A10                       /AUTO INDEX REGISTER 10
0351  1273                  TAD MODE                      / 0 IF 12-BIT, 100 IF 8 BIT
0352  1274                  TAD FUNCUN                    / 2 MEANS EMPTY BUFFER
0353  1275                  TAD     DRIVEP                /GET DENSITY
0354  3276                  DCA     COMMAND               /STORE ASSEMBLED COMMAND
0355  1276                  TAD     COMMAND               /GET COMMAND TO AC
0356  6731                  LCD                           /ISSUE COMMAND TO RX
0357  1276                  TAD     COMMAND               /GET SAVED COMMAND
0360  0267                  AND     K100                  /MASK FOR 8-BIT MODE
0361  7650                  SNA CLA                       /IF 8-BIT MODE SET DO 8-MODE PROTOCOL
0362  5776'                 JMP     ELOOP                 /IF 12-BIT MODE GO STRAIGHT TO EMPTY LOOP
0363  1276                  TAD     COMMAND               /GET SAVED COMMAND WORD
0364  7106                  CLL RTL                       /GET 4 MSB'S (BITS 0,1,2,3) OF
0365  7006                  RTL                           /COMMAND WORD DOWN TO THE
0366  7004                  RAL                           /4 LSB'S (BITS 8,9,10,11) OF AC
0367  6733                  STR                           /IOT TO SKIP ON TRANSFER READY
0370  5367                  JMP     .-1                   /IF TR NOT SET LOOP UNTIL IT DOES
0371  6732                  XDR                           /ISSUE SECOND COMMAND WORD
0372  7200                  CLA                           /CLEAR THE AC AND DO EMPTY LOOP
0373  5776'                 JMP     ELOOP                 /GET OVER TO NEXT PAGE
0376  0400
0377  0577
      0400          PAGE
                    /
                    /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
                    /
0400  6733          ELOOP,  STR                           /TEST FOR TR FLAG
0401  7410                  SKP                           /TR NOT SET, TEST FOR DONE FLAG
0402  5212                  JMP EMPTY                     /TR FLAG SET
0403  6735                  SDN                           /TEST FOR DONE FLAG
0404  5200                  JMP ELOOP                     /NOT TR, OR DONE YET
                    /
                    /THE DONE FLAG IS SET
                    /
                    /TEST FOR ANY ERRORS
                    /
0405  6734                  SER                           /TEST FOR THE ERROR FLAG
0406  7402                  HLT                           /NO ERRORS - OK
                    /
                    /INCREMENT AND TEST THE ERROR RETRY PROGRAM LOCATION "ETRY"
                    /
                    /AND RETRY THE COMMAND UNTIL THE ERROR RECOVERS
                    /
                    /OR UNTIL THE ETRY COUNTER OVERFLOWS TO 0
                    /
0407  2777'                 ISZ ETRY
0410  5776'                 JMP ESETUP                    /RETRY TO EMPTY THE SECTOR BUFFER
0411  7402                  HLT                           /HARD ERROR
                    /
                    /THE TRANSFER REQUEST FLAG IS SET
                    /
                    /TRANSFER DATA TO THE PROGRAMS DATA BUFFER FROM THE RX01 SECTOR BUFFER
                    /
0412  6732          EMPTY,  XDR                           /FROM THE RX01 SECTOR BUFFER
0413  3410                  DCA I A10                     /TO THE PROGRAMS DATA BUFFER
0414  5200                  JMP ELOOP                     /LOOP UNTIL THE DONE FLAG SETS

0576  0347
0577  0272
      0600                  PAGE
                    /THE FOLLOWING PROGRAM LOCATIONS ARE RESERVED FOR THE PROGRAMS DATA BUFFER
                    /
0600  0000          BUFFER, 0
      1200                  *BUFFER+400
                            $
```

Figure 4-16 RX28 Empty Buffer Example

### 4.1.8 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.

   ```
   STR
   SKP
   JMP
   SDN
   JMP
   (done)
   (fill or empty buffer)
   ```

   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 4.1.4.9), 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.

## 4.2 RX11 AND RXV11 PROGRAMMING INFORMATION

This section describes device registers, register and vector address assignments, programming specifications, and programming examples for the RX11 and RXV11 interfaces.

All software control of the RX11/RXV11 is performed by means of two device registers: the command and status (RXCS) register and a multipurpose data buffer (RXDB) register. These registers have been assigned bus addresses (Paragraph 4.2.1) and can be read or loaded, with certain exceptions, using any instruction referring to their addresses.

The RX02, 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 RX02.

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 4.2.2.1) 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 RX02. The program retests TR and moves another byte of data when TR is true. When the RX02 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 RX02 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 RX02 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.

### 4.2.1 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.

### 4.2.2 Register Description

**4.2.2.1 RXCS – Command and Status (177170)** – Loading this register while the RX02 is not busy and with bit 0=1 will initiate a function as described below and indicated in Figure 4-17. Bits 0–4 are write-only bits.

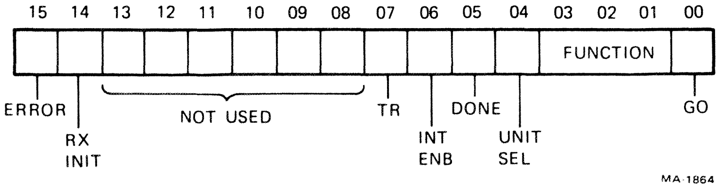


Figure 4-17 RXCS Format (RX11, RXV11)

**Bit No. Description**

0 Go – Initiates a command to RX02. This is a write-only bit.

1–3 Function Select – These bits code one of the eight possible functions listed below and described in Paragraph 4.2.3. 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.

5 Done – This bit indicates the completion of a function. Done will generate an interrupt when asserted if Interrupt Enable (RX2CS 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 RX02 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 or RXV11 needs data or has data available. This is a read-only bit.

8–13 Unused

14 RX Initialize – This bit is set by the program to initialize the RX11 or RXV11 without initializing all 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 or RXV11 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 RX02 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 RX02 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 4.2.6).

**4.2.2.2 RXDB – Data Buffer Register (177172)** – This register serves as a general purpose data path between the RX02 and the interface. It may represent one of four RX02 registers according to the protocol of the function in progress (Paragraph 4.2.3).

This register is read/write if the RX02 is not in the process of executing a command; that is, it may be manipulated without affecting the RX02 subsystem. If the RX02 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.**

**4.2.2.3 RXTA – RX Track Address (Figure 4-18)** – This register is loaded to indicate on which of the 77 ($114_8$) tracks a given function is to operate. It can be addressed only under the protocol of the function in progress (Paragraph 4.2.3). Bits 8–15 are unused and are ignored by the control.

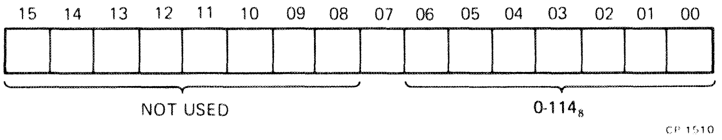


Figure 4-18 RXTA Format (RX11/RXV11)

**4.2.2.4 RXSA – RX Sector Address (Figure 4-19)** – This register is loaded to indicate on which of the 26 ($32_8$) sectors a given function is to operate. It can be addressed only under the protocol of the function in progress (Paragraph 4.2.3). Bits 8–15 are unused and are ignored by the control.

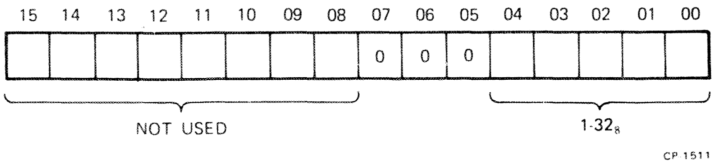


Figure 4-19 RXSA Format (RX11/RXV11)

**4.2.2.5 RXDB – RX Data Buffer (Figure 4-20)** – 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 4.2.3).

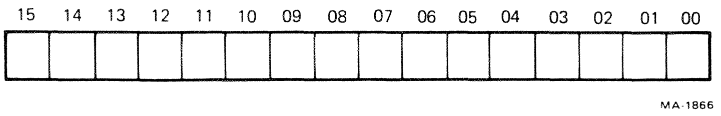


Figure 4-20 RXDB Format (RX11/RXV11)

**4.2.2.6 RXES - RX Error and Status (Figure 4-21)** - 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 4.2.3). The RXES content is located in the RXDB upon completion of a function.

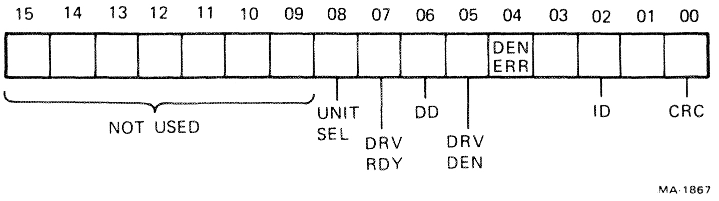


Figure 4-21 RXES Format (RX11, RXV11)

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.

2 Initialize Done - This bit is asserted in the RXES to indicate completion of the Initialize routine which can be caused by RX02 power failure, system power failure, or programmable or Unibus Initialize.

3

4 Density Error - This bit is asserted to indicate the density of the function in progress does not match the drive density. Upon detection of this error the control terminates the operation and Error and Done are asserted.

NOTE

**Bits 4 and 5 are asserted for the occurrence of double density when the system is RX01-compatible.**

5 Drive Density - This bit indicates the density of the diskette in the drive selected. When asserted, double density is indicated.

6 Deleted Data Detected - During data recovery, the identification mark preceding the data field was decoded as a deleted data mark (Paragraph 1.5.3.2).

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, specific error conditions can be accessed via a read error register function (Paragraph 4.2.3.7).**

8 Unit Select - Drive 0 is selected if this bit is "0"; drive 1 is selected if this bit is a "1."

### 4.2.3 Function Codes

Following the strict protocol of the individual function, data storage and recovery on the RX11 and RXV11 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.

**4.2.3.1 Fill Buffer (000)** - This function is used to fill the RX02 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, RXES, OUT, and Done are cleared. When the TR bit is asserted, the first byte of the data may be loaded into the data buffer. The control then clears TR and after supplying the appropriate number of shift pulses to store the data, again asserts TR. The same TR cycle will occur as each byte of data is loaded. The RX02 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/RXV11. When the complete buffer has been filled, the control asserts Done.

**4.2.3.2 Empty Buffer (001)** - This function is used to empty into the interface the 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. For this function, TR and shift pulses are generated in the same manner as for the fill buffer but the buffer is emptied.

When TR sets, the program may unload the first of 128 data bytes from the RXDB. Then the RX11/RXV11 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.

**NOTE**
**The empty buffer function does not destroy the contents of the sector buffer.**

**4.2.3.3 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 TR and 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/RXV11 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 RX02 attempts to locate the desired sector. If the RX02 is unable to locate the desired sector within two diskette revolutions, the RX11/RXV11 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/RXV11 will write the 128 bytes stored in the internal buffer followed by a 16-bit CRC character that is automatically calculated by the RX02. The RX11/RXV11 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 RX02. 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.**

**4.2.3.4 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 RXES, Done, and OUT.

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/RXV11 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 RX02 attempts to locate the desired track and sector. If the RX02 is unable to locate the desired sector within two diskette revolutions after locating the presumably correct track, the RX11/RXV11 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/RXV11 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.

**4.2.3.5 Read Status (101)** – The RX11/RXV11 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 read status 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 4.2.2.6.

NOTE

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

**4.2.3.6 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 1.5.3.2).

**4.2.3.7 Read Error Code Function (111)** – The read error code 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 4.2.6).

NOTE

**Care should be exercised in the use of this function, since under certain conditions, erroneous error information may result (Paragraph 4.2.5).**

**4.2.3.8 Power Fail** – There is no actual function code associated with Power Fail. When the RX02 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 RX02 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 2 (Initialize Done) (Paragraph 4.2.2.6) 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 (RX 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.

### 4.2.4 Programming Examples

**4.2.4.1 Read Data/Write Data** - Figure 4-22 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/RXV11 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/RXV11 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 a parity error has occurred, a branch is made to PARER. If neither of the above occurs, 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 10 times by decrementing the respective retry counter. If an error persists after 10 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.**

**4.2.4.2 Empty Buffer Function** - Figure 4-23 shows a program for implementing an empty buffer function. The first instruction sets the number of error retries to 10. 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 10 retries, the error is considered hard.

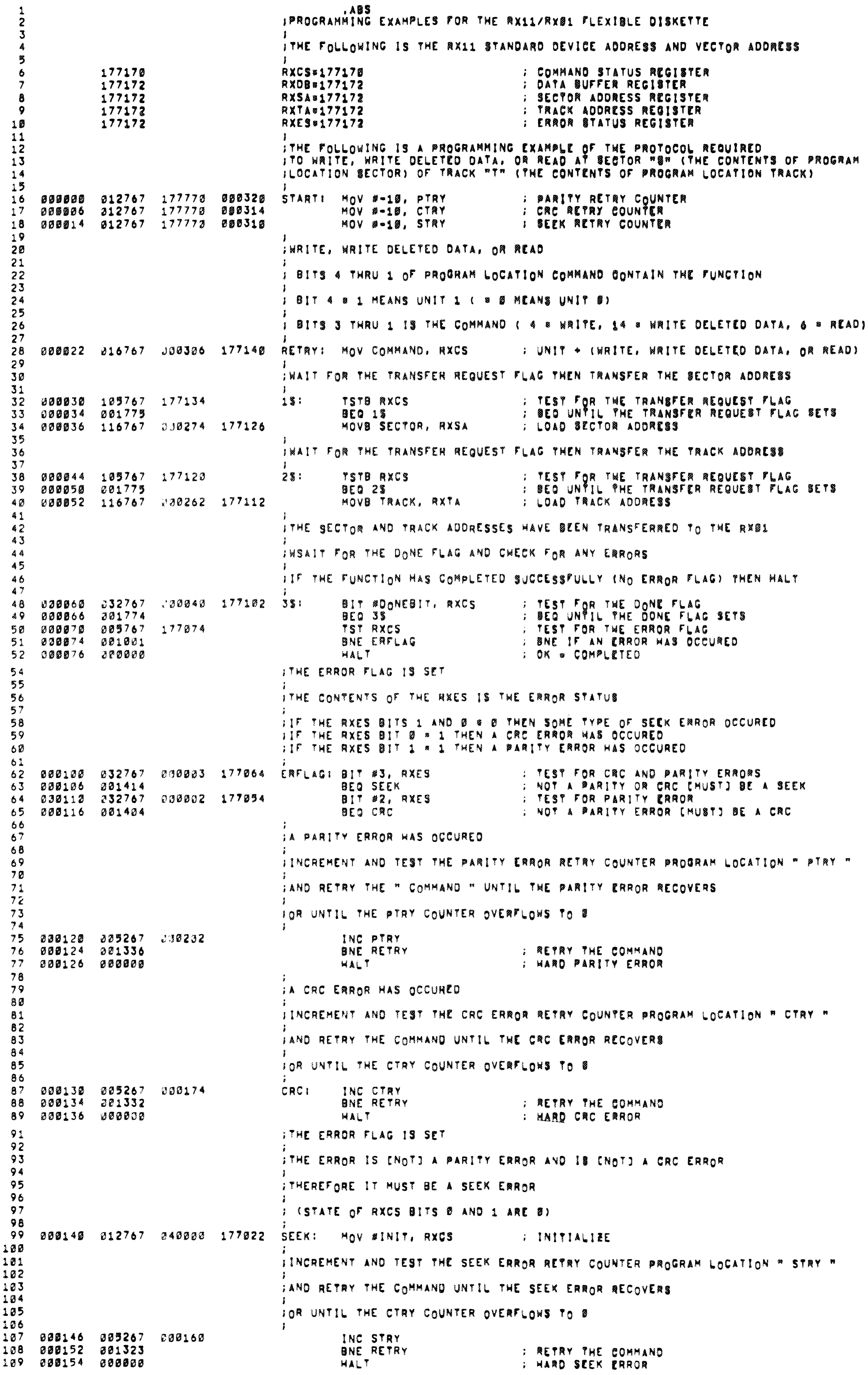
If no error is indicated, the program looks for the transfer request (TR) flag to set. The error flag is retested if TR is not set. Once TR sets, a byte is moved from the RX11/RXV11 sector buffer to the core locations of BUFFER. The process continues until the sector buffer is empty and the Done bit is set.

**4.2.4.3 Fill Buffer Function** - Figure 4-24 presents a program to implement a fill buffer function. It is very similar to the empty buffer example.

### 4.2.5 Restrictions and Programming Pitfalls

A set of restrictions and programming pitfalls for the RX11/RXV11 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.

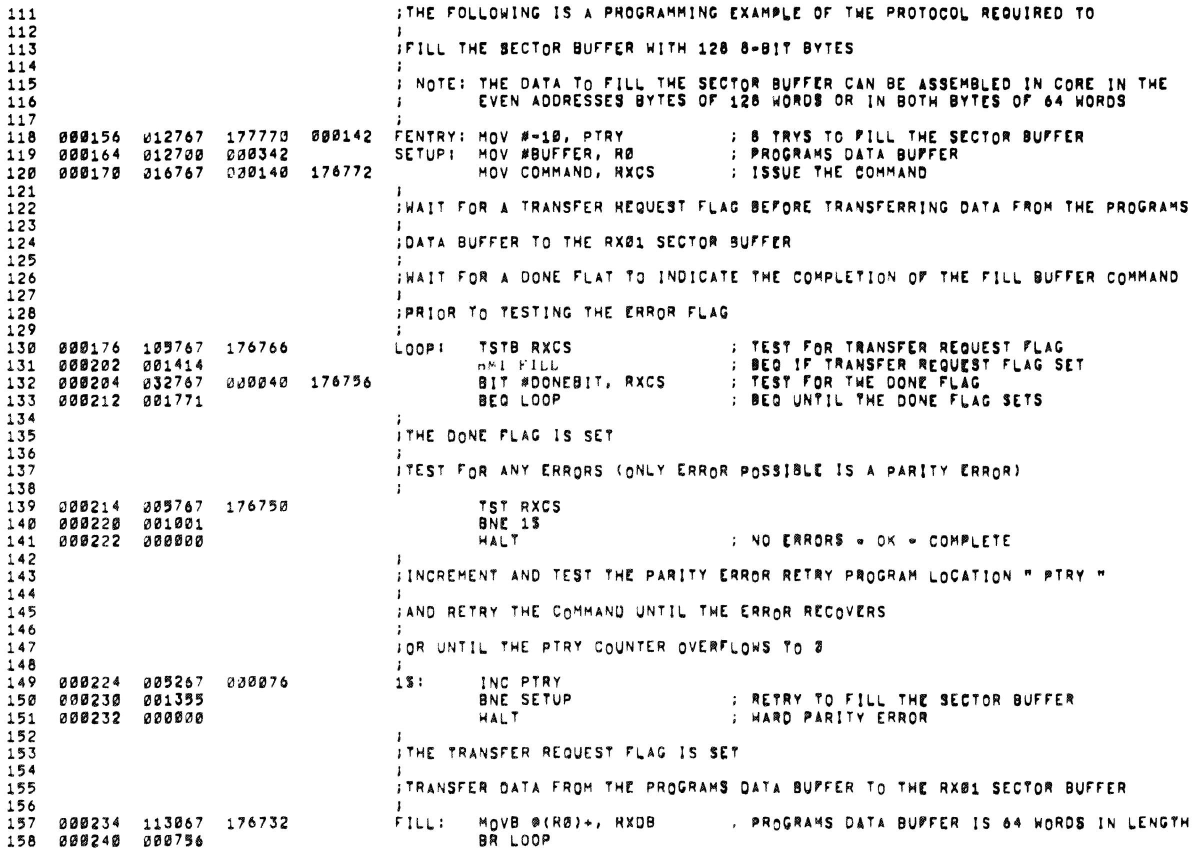


```
                                .ABS
                                ;PROGRAMMING EXAMPLES FOR THE RX11/RX01 FLEXIBLE DISKETTE
                                ;
                                ;THE FOLLOWING IS THE RX11 STANDARD DEVICE ADDRESS AND VECTOR ADDRESS
                                ;
        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  000320  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
                                ;
                                ;WSAIT 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  001001                          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
000112  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  000232                  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  000160                  INC STRY
000152  001323                          BNE RETRY               ; RETRY THE COMMAND
000154  000000                          HALT                    ; HARD SEEK ERROR
```

Figure 4-22 RX11/RXV11 Write/Write Deleted Data/Read Example

```
                                     ;THE FOLLOWING IS A PROGRAMMING EXAMPLE OF PROTOCOL REQUIRED TO
                                     ;
                                     ;EMPTY THE  SECTOR BUFFER OF 128 8-BIT BYTES
                                     ;
000242  012767  177770  000056  EENTRY: MOV #-10, PTRY          ; 8 TRYS TO EMPTY THE SECTOR BUFFER
000250  012700  000342          ESETUP: MOV #BUFFER, R0         ; PROHGRAMS DATA BUFFER
000254  016767  000054  176706          MOV COMMAND, RXCS       ; ISSUE THE COMMAND
                                ;
                                ;WAIT FOR A TRANSFER REQUEST FLAG BEFORE TRANSFERRING DATA TO THE PROGRAMS
                                ;
                                ;DATA BUFFER FRON THE RX01 SECTOR BUFFER
                                ;
                                ;WAIT FOR A DONE FLAG TO INDICATE THE COMPLETION OF THE EMPTY BUFFER COMMAND
                                ;
                                ;PRIOR TO TESTIN G THE ERROR FLAG
                                ;
000262  105767  176702          ELOOP:  TSTB RXCS               ; TEST FOR TRANSFER REQUEST FLAG
000266  001014                          BMI EMPTY               ; BNE IF TRANSFER REQUEST FLAG IS SET
000270  032767  000040  176672          BIT #DONEBIT, RXCS      ; TEST FOR DONE FLAG
000276  001771                          BEQ ELOOP               ; BEQ UNTIL THE DONE FLAG SETS
                                ;
                                ;THE DONE FLAG IS SET
                                ;
                                ;TEST FOR ANY ERRORS (ONLY ERROR POSSIBLE IS A PARITY ERROR)
                                ;
000300  005767  176664                  TST RXCS
000304  001001                          BNE 1$
000306  000000                          HALT                    ; NO ERRORS = OK = COMPLETE
                                ;
                                ;INCFEMENT AND TEST THE PARITY ERROR RETRY PROGRAM LOCATION " PTRY "
                                ;
                                ;AND RETRY THE COMMAND UNTIL THE ERROR RECOVERS
                                ;
                                ;OR UNTIL THE PTRY CUNTER OVERFLOWS TO 0
                                ;
000310  005267  000012          1$:     INC PTRY
000314  001355                          BNE ESETUP              ; RETRY TO EMPTY THE SECTOR BUFFER
000316  000000                          HALT                    ; HARD PARITY ERROR
                                ;
                                ;THE TRANSFER REQUEST FLAG IS SET
                                ;
                                ;TRANSFER DATA TO THE PHOGRAM  DATA BUFFER FROM THE RX01 SECTOR BUFFER
                                ;
000320  116730  176646          EMPTY:  MOVB RXDB, @(R0)+
000324  000756                          BR ELOOP

                                ;THE FOLLOWING 3 PROGHAM LOCATIONS ARE THE ERROR RETRY COUNTERS
                                ;
000326  000000                  PTRY:   0                       ; PARITY ERROR RETRY COUNTER
000330  000000                  CTRY:   0                       ; CRC ERROR RETRY COUNTER
000332  000000                  STRY:   0                       ; SEEK ERROR RETRY COUNTER
                                ;
                                ;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)
                                ;
000334  000000                  COMMAND:        0               ; 4, 14, 6, OR 2 + (GO BIT 1 = 1)
                                ;
                                ;PROGRAM LOCATION " SECTOR " CONTAINS THE SECTOR ADDRESS (1 TO 32 OCTAL)
                                ;
000336  000000                  SECTOR: 0                       ; 1 TO 32 OCTAL
                                ;
                                ;PROGRAM LOCATION " THACK " CONTAINS THE TRACK ADDRESS (0 TO 114 OCTAL)
                                ;
000340  000000                  TRACK:  0                       ; 0 TO 114 OCTAL
                                ;
                                ;PROGRAM EQUIVALENTS
                                ;
        000040                  DONEBIT=40
        040000                  INIT=40000
        000342                  BUFFER=.
        000542                          .=BUFFER+200
        000001                          .END
```

Figure 4-23 RX11/RXV11 Empty Buffer Example



```
                                   ;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     FENTRY: 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  113067  176732             FILL:   MOVB @(R0)+, RXDB       . PROGRAMS DATA BUFFER IS 64 WORDS IN LENGTH
000240  000756                             BR LOOP
```

Figure 4-24 RX11/RXV11 Fill Buffer Example

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

3. The DRV SEL RDY bit is only updated at the time of a read status function (101) for both drives, and after an Initialize, depending on the status of drive 0. At the termination of any other functions it reflects the drive status of the last Read Status or Initialize command.

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 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.

### 4.2.6 Error Recovery

There are two error indications given by the RX11/RXV11 system. The read status function (Paragraph 4.2.3.5) will assemble the current contents of the RXES (Paragraph 4.2.2.6), which can be sampled to determine errors. The read error code function (Paragraph 4.2.3.7) can also be used to retrieve explicit error information. The RX11/RXV11 interface register can be interrogated to determine the type of failure that occurred. A list of error codes follows.

> **NOTE**
> **A read status function is not necessary if the DRV RDY bit is not going to be interrogated because the RX2ES 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 |
| 0040 | Tried to access a track greater than 77 |
| 0050 | Home was found before desired track was reached |
| 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 ID mark found within allowable time span |
| 0140 | CRC error on what appeared to be a header. Error is not asserted |
| 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) |
| 0200 | CRC error on reading the sector from the disk |
| 0220 | R/W electronics failed maintenance mode test |
| 0240 | Density Error |

## 4.3 RX211 AND RXV21 PROGRAMMING INFORMATION

This section describes device registers, register and vector address assignments, programming specifications, and programming examples for the RX211 and RXV21 interfaces.

All software control of the RX211/RXV21 is performed by means of two device registers: the command and status register (RX2CS) and a multipurpose data buffer register (RX2DB) which have been assigned bus addresses and can be read or loaded.

The RX02 contains all the control circuitry required to read from and write on the disk and to calculate and verify the CRC. It has a buffer large enough to hold one full sector of diskette data (128 or 256 8-bit bytes). Information is serially passed between the interface and the RX02.

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

**Fill Buffer** – A command to fill the buffer is moved into the RX2CS. The Go bit must be set. The program tests for TR. When TR is detected, the program moves the desired word count into the RX2DB. TR goes false while the word count is moved to the RX02. The program retests TR and moves the bus address into the RX2DB. The device now requests bus mastership and DMA's one data word at a time into the RX2DB and shifts it across the RX02 data bus serially one 8-bit byte at a time into the sector buffer. When the word count register overflows (if necessary, the RX02 control zero-fills the remainder of the sector buffer) the Done bit is set, and an interrupt will occur if the program has enabled interrupts.

**Write Sector** – A command to write the contents of the sector buffer onto the disk is moved into the RX2CS. The program tests TR and when TR is set, moves the desired sector address to the RX2DB. TR remains false while the sector address is shifted to the RX02 control. The control retests TR and when it is again set, moves the desired track address register to the RX2DB. Again TR is negated. The RX02 locates the desired track and sector and compares the diskette density against the assigned function density and writes the contents of the sector buffer onto the disk if the densities agree. When the write operation is completed, the Done bit is set and an interrupt will occur if the program has enabled interrupts.

A typical disk read operation occurs in the reverse order. First, the desired track and sector are located and the contents of the sector are read into the sector buffer (read sector). Then the contents of the sector buffer is unloaded into memory (empty buffer). In either case, the contents of the sector buffer are not valid if either a Power Fail or Initialize follows a fill buffer or read sector function.

### 4.3.1 Register and Vector Addresses

The RX211/RXV21 use two registers for communicating with the host computer: the command and status register (RX2CS) normally assigned bus address 177170 and the data buffer register (RX2DB) normally assigned bus address 177172. The vector address is 264.

### 4.3.2 Register Description

**4.3.2.1 RX2CS – Command and Status (177170)** – Loading this register while the RX02 is not busy and with bit 0=1 will initiate a function as described below and indicated in Figure 4-25.

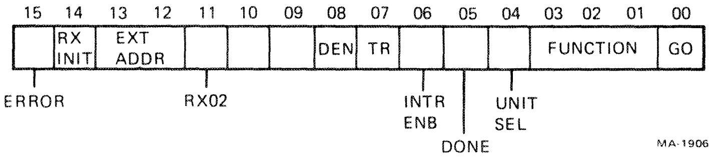


Figure 4-25 RX2CS Format RX211/RXV21

**Bit No.** **Description**

0 Go – Initiates a command to RX02. This is a write-only bit.

1-3 Function Select – These bits code one of the eight possible functions described in Paragraph 4.3.3 and listed below. These are write-only bits.

| **Code** | **Function** |
|---|---|
| 000 | Fill Buffer |
| 001 | Empty Buffer |
| 010 | Write Sector |
| 011 | Read Sector |
| 100 | Set Media Density |
| 101 | Read Status |
| 110 | Write Deleted Data Sector |
| 111 | Read Error Code |

4 Unit select – This bit selects one of the two possible disks for execution of the desired function. This bit is readable only when Done is set, at which time it indicates the unit previously selected. This is a read/write bit.

5 Done - This bit indicates the completion of a function. Done will generate an interrupt when asserted if Interrupt Enable (RX2CS 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 RX02 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 RX211/RXV21 needs data or has data available. This is a read-only bit.

8 Density - This bit determines the density of the function to be executed. This bit is readable only when Done is set, at which time it indicates the density of the function previously executed. This is a read/write bit.

9-10 Reserved for future use. Must be written as a zero.

11 RX02 - This bit is set by the interface to inform the programmer that this is an RX02 system. This is a read-only bit.

12-13 Extended address - These bits are used to declare an extended bus address. These are write-only bits.

14 RX211/RXV21 Initialize - This bit is set by the program to initialize the RX211/RXV21 without initializing all devices on the Unibus. This is a write-only bit.

CAUTION

**Loading the lower byte of the RX2CS will also load the upper byte of the RX2CS.**

Upon setting this bit in the RX2CS, the RX211/RXV21 will negate Done and move the head position mechanism of both drives (if two are available) to track 0. Upon completion of a successful Initialize, the RX02 will zero the error and status register, and set Initialize Done. It will also read sector 1 of track 1 on drive 0 into the buffer.

15 Error - This bit is set by the RX02 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.

**4.3.2.2 RX2DB - Data Buffer Register** (177172) - This register serves as a general purpose data path between the RX02 and the interface. It may represent one of six RX02 registers according to the protocol of the function in progress (Paragraph 4.3.3).

This register is read/write if the RX02 is not in the process of executing a command; that is, it may be manipulated without affecting the RX02 subsystem. If the RX02 is actively executing a command, this register will only accept data if RX2CS 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.**

**4.3.2.3 RX2TA - RX Track Address** (Figure 4-26) - This register is loaded to indicate on which of the $114_8$ (0–$76_{10}$) tracks a given function is to operate. It can be addressed only under the protocol of the function in progress (Paragraph 4.3.3). Bits 8–15 are unused and are ignored by the control.

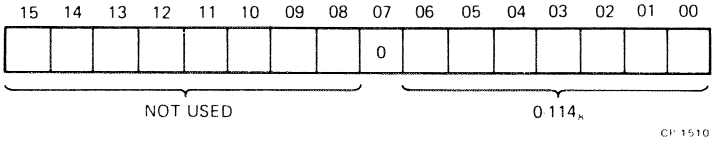


Figure 4-26 RX2TA Format (RX211/RXV21)

**4.3.2.4 RX2SA - RX Sector Address** (Figure 4-27) - This register is loaded to indicate on which of the $32_8$ (1-$26_{10}$) sectors a given function is to operate. It can be addressed only under the protocol of the function in progress (Paragraph 4.3.3).

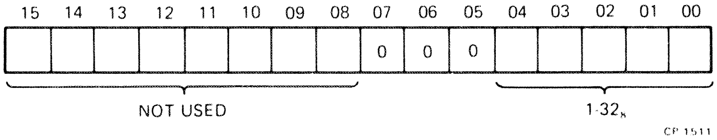


Figure 4-27 RX2SA Format (RX211/RXV21)

**4.3.2.5 RX2WC - RX Word Count Register** (Figure 4-28) - For a double density sector the maximum word count is $128_{10}$. For a single density sector the maximum word count is $64_{10}$. If a word count is beyond the limit for the density indicated, the control asserts Word Count Overflow (bit 10 of RX2ES). This is a write-only register. The actual word count and not the 2's complement of the word count is loaded into the register.

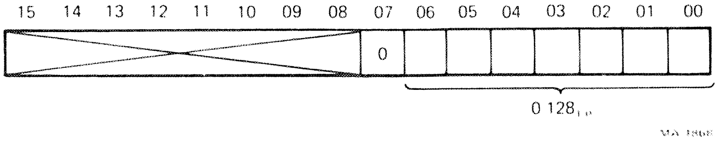


Figure 4-28 RX2WC Format (RX211/RXV21)

**4.3.2.6 RX2BA - RX Bus Address Register** (Figure 4-29) - This register specifies the bus address of data transferred during fill buffer, empty buffer, and read definitive error operations. Incrementation takes place after a memory transaction has occurred. The RX2BA, therefore, is loaded with the ad dress of the first data word to be transferred. This is a 16-bit, write-only register (Paragraph 4.3.3).

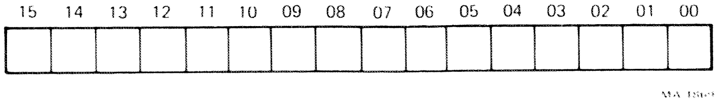


Figure 4-29 RX2BA and RX2DB Format (RX211/RXV21)

**4.3.2.7 RX2DB - RX Data Buffer** (Figure 4-29) - 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 4.3.3).

**4.3.2.8 RX2ES - RX Error and Status** (Figure 4-30) - This register contains the current error and status conditions of the drive selected by bit 4 (Unit Select) of the RX2CS. This read-only register can be addressed only under the protocol of the function in progress (Paragraph 4.3.3). The RX2ES is located in the RX2DB upon completion of a function.

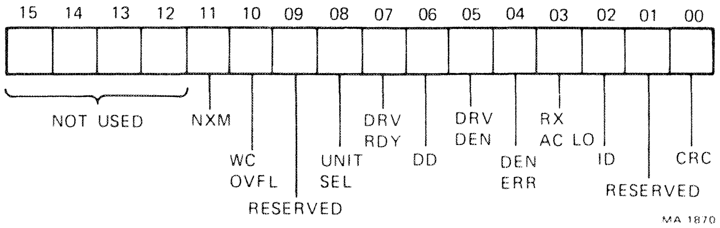


Figure 4-30 RX2ES Format (RX211/RXV21)

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 data collected must be considered invalid. The RX2ES is moved to the RX2DB, and Error and Done are asserted. It is suggested that the data transfer be retried up to 10 times, as most errors are recoverable (soft).

2 Initialize Done - This bit is asserted in the RX2ES to indicate completion of the Initialize routine which can be caused by RX02 power failure, system power failure, or programmable or bus Initialize.

3 RX AC LO - This bit is set by the interface to indicate a power failure in the RX02 subsystem.

4 Density Error - This bit indicates that the density of the function in progress does not match the drive density. Upon detection of this error the control terminates the operation and asserts Error and Done.

5 Drive Density - This bit indicates the density of the diskette in the drive selected (indicated by bit 8). The density of the drive is determined during read and write sector operations.

6 Deleted Data - This bit indicates that in the course of recovering data, the "deleted data" address mark was detected at the beginning of the data field. The Drv Den bit indicates whether the mark was a single or double density deleted data address mark. The data following the mark will be collected and transferred normally, as the deleted data mark has no further significance other than to establish drive density. Any alteration of files or actual deletion of data due to this mark must be accomplished by user software.

7 Drive Ready - This bit indicates that the selected drive is ready if bit 7=1 and all conditions for disk operation are satisfied, such as door closed, power okay, diskette up to speed, etc. The RX02 may be presumed to be ready to perform any operation. This bit is only valid when retrieved via a read status function or initialize.

8 Unit Select - This bit indicates that drive 0 is selected if bit 8=0. This bit indicates the drive that is currently selected.

10 Word Count Overflow - This bit indicates that the word count is beyond sector size. The fill or empty buffer operation is terminated and Error and Done are set.

11 Nonexistent Memory Error - This bit is set by the interface when a DMA transfer is being performed and the memory address specified in RX2BA is nonexistent.

### 4.3.3 Function Codes

Following the strict protocol of the individual function, data storage and recovery on the RX211/RXV21 occur with careful manipulation of the RX2CS and RX2DB 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 | Set Media Density |
| 101 | Read Status |
| 110 | Write Deleted Data Sector |
| 111 | Read Error Code |

The following paragraphs describe in detail the programming protocol associated with each function encoded and written into RX2CS bits 1-3 if Done is set.

**4.3.3.1 Fill Buffer (000)** - This function is used to fill the RX02 data buffer with the number of words of data specified by the RX2WC register. Fill buffer is a complete function in itself: the function ends when RX2WC overflows, and if necessary, the control has zero-filled the remainder of the buffer. The contents of the buffer may be written on the disk by means of a subsequent Write Sector command or returned to the host processor by an Empty Buffer command. If the word count is too large, the function is terminated, Error and Done are asserted, and the Word Count overflow bit is set in RX2ES.

To initiate this function the RX2CS is loaded with the function. Bit 4 of the RX2CS (Unit Select) does not affect this function since no disk operation is involved. Bit 8 (Density) must be properly selected since this determines the word count limit. When the command has been loaded, the Done bit (RX2CS bit 5) goes false. When the TR bit is asserted the RX2WC may be loaded into the data buffer register. When TR is again asserted, the RX2BA may be loaded into the RX2DB. The data words are transferred directly from memory and when RX2WC overflows and the control has zero-filled the remainder of the sector buffer, if necessary, Done is asserted ending the operation. If bit 6 RX2CS (Interrupt Enable) is set, an interrupt is initiated. Any read of the RX2DB during the data transfer is ignored by the interface. After Done is true the RX2ES is located in the RX2DB register.

**4.3.3.2 Empty Buffer (001)** - This function is used to empty the contents of the internal buffer through the RX211/RXV21 for use by the host processor. This data is in the buffer as the result of a previous Fill Buffer or Read Sector command.

The programming protocol for this function is identical to that for the Fill Buffer command. The RX2CS is loaded with the command to initiate the function. (This function will ignore bit 4 RX2CS, Unit Select). RX2CS bit 8 (Density) must be selected to allow the proper word count limit. When the command has been loaded, the Done bit (RX2CS bit 5) goes false. When the TR bit is asserted, the RX2WC may be loaded into the RX2DB. When TR is again asserted the RX2BA may be loaded into the RX2DB. The RX211/RXV21 assembles one word of data at a time and transfers it directly to memory. Transfers occur until word count overflow, at which time the operation is complete and Done goes true. If bit 6 RX2CS (Interrupt Enable) is set, an interrupt is initiated. After Done is true, the RX2ES is located in the data buffer register.

**4.3.3.3 Write Sector (010)** - This function is used to locate a desired sector on the diskette and fill it with the contents of the internal buffer. The initiation of the function clears RX2ES, TR, and Done.

When TR is asserted, the program must load the desired sector address into RX2DB, which will drop TR. When TR is again asserted, the program must load the desired track address into the RX2DB, which will drop TR. TR will remain unasserted while the RX02 attempts to locate the desired sector. The diskette density is determined at this time and is compared to the function density. If the densities do not agree, the operation is terminated; bit 4 RX2ES is set, RX2ES is moved to the RX2DB, Error (bit 15 RX2CS) is set, Done is asserted, and an interrupt is initiated, if bit 6 RX2CS (Interrupt Enable) is set.

If the densities agree but the RX02 is unable to locate the desired sector within two diskette revolutions, the interface will abort the operation, move the contents of RX2ES to the RX2DB, set Error (bit 15 RX2CS), assert Done, and initiate an interrupt if bit 6 RX2CS (Interrupt Enable) is set.

If the desired sector has been reached and the densities agree, the RX211/RXV21 will write the $128_{10}$ or $64_{10}$ words stored in the internal buffer followed by a CRC character which is automatically calculated by the RX02. The RX211/RXV21 ends the function by asserting Done and if bit 6 RX2CS (Interrupt Enable) is set, initiating an interrupt.

CAUTION

**The contents of the sector buffer are not valid data after a power loss has been detected by the RX02. However, write sector will be accepted as a valid instruction and the (random) contents of the buffer will be written, followed by a valid CRC.**

NOTE

**The contents of the sector buffer are not destroyed during a write sector operation.**

**4.3.3.4 Read Sector (011)** - This function is used to locate the desired sector and transfer the contents of the data field to the internal buffer in the control. This function may also be used to retrieve rapidly (5 ms) the current status of the drive selected. The initiation of this function clears RX2ES, TR, and Done.

When TR is asserted the program must load the desired sector address into the RX2DB, which will drop TR. When TR is again asserted, the program must load the desired track address into the RX2DB, which will drop TR.

TR and Done will remain negated while the RX02 attempts to locate the desired sector. If the RX02 is unable to locate the desired sector within two diskette revolutions for any reason, the RXV21/RX211 will abort the operation, set Done and Error (bit 15 RX2CS), move the contents of the RX2ES to the RX2DB, and if bit 6 RX2CS (Interrupt Enable) is set, initiate an interrupt.

If the desired sector is successfully located, the control reads the data address mark and determines the density of the diskette. If the diskette (drive) density does not agree with the function density the operation is terminated and Done and Error (bit 15 RX2CS) are asserted. Bit 4 RX2ES is set (Density Error) and the RX2ES is moved to the RX2DB. If bit 6 RX2CS (Interrupt Enable) is set, an interrupt is initiated.

If a legal data mark is successfully located, and the control and densities agree, the control will read data from the sector into the internal buffer. If a deleted data address mark was detected, the control will set bit 6 RX2ES (DD). As data enters the internal buffer, a CRC is computed based on the data field and the CRC bytes previously recorded. A non-zero residue indicates that a read error has occurred and the control sets bit 0 RX2ES (CRC error) and bit 15 RX2CS (Error). The RX211/RXV21 ends the operation by asserting Done and moving the contents of the RX2ES into the RX2DB. If bit 6 RX2CS is set, an interrupt is initiated.

If the desired sector is successfully located, the densities agree, and the data is transferred with no CRC error, Done will be set and if bit 6 RX2CS (Interrupt Enable) is set the RX211/RXV21 initiates an interrupt.

**4.3.3.5 Set Media Density (100)** – This function causes the entire diskette to be reassigned to a new density. Bit 8 RX2CS (Density) indicates the new density. The control reformats the diskette by writing new data address marks (double or single density) and zeroing all of the data fields on the diskette.

The function is initiated by loading the RX2CS with the command. Initiation of the function clears RX2ES and Done. When TR is set, an ASCII "I" (111) must be loaded into the RX2DB to complete the protocol. This extra character is a safeguard against an error in loading the command. When the control recognizes this character it begins executing the command.

The control starts at sector 1, track 0 and reads the header information, then starts a write operation. If the header information is damaged, the control will abort the operation.

If the operation is successfully completed, Done is set and if bit 6 RX2CS (Interrupt Enable) is set an interrupt is initiated.

CAUTION

**This operation takes about 15 seconds and should not be interrupted. If for any reason the operation is interrupted, an illegal diskette has been generated which may have data marks of both densities. This diskette should again be completely reformatted.**

**4.3.3.6 Maintenance Read Status (101)** – This function is initiated by loading the RX2CS with the command. Done is cleared. The Drive Ready bit (bit 7 RX2ES) is updated by counting index pulses in the control. The Drive Density is updated by loading the head of the selected drive and reading the first data mark. The RX2ES is moved into the RX2DB. The RX2CS may be sampled when Done (bit 5 RX2CS) is again asserted and if bit RX2CS (Interrupt Enable) is set, an interrupt will occur. This operation requires approximately 250 ms to complete.

**4.3.3.7 Write Sector with Deleted Data (110)** – This operation is identical to function 010 (write sector) with the exception that a deleted data address mark is written preceding the data rather than the standard data address mark. The Density bit associated with the function indicates whether a single or double density deleted data address mark will be written.

**4.3.3.8 Read Error Code (111)** – The read error code function implies a read extended status. In addition to the specific error code a dump of the control's internal scratch pad registers also occurs. This is the only way that the word count register can be retrieved. This function is used to retrieve specific information as well as drive status information depending upon detection of the general Error bit.

The transfer of the registers is a DMA transfer. The function is initiated by loading the RX2CS with the command and then Done goes false. When TR is true, the RX2BA may be loaded into the RX2DB and TR goes false. The registers are assembled one word at a time and transferred directly to memory.

**Register Protocol**

| | | |
|---|---|---|
| Word 1<7:0> | Definitive Error Codes | |
| Word 1<15:8> | Word Count Register | |
| Word 2<7:0> | Current Track Address of Drive 0 | |
| Word 2<15:8> | Current Track Address of Drive 1 | |
| Word 3<7:0> | Target Track of Current Disk Access | |
| Word 3<15:8> | Target Sector of Current Disk Access | |
| Word 4<7> | Unit Select Bit | * |
| Word 4<5> | Head Load Bit | * |
| Word 4<6><4> | Drive Density Bit of Both Drives | * |
| Word 4<0> | Density of Read Error Register Command | * |
| Word 4<15:8> | Track Address of Selected Drive | † |

* For DMA interfaces the controller status soft register is sent to the interface at the end of the command. The four status bits are included in an 8-bit word. Unit Select = bit 7, Density of Drive 1 = bit 6, Head Load = bit 5, Density of Drive 0 = bit 4, Density of Read Error Register Command = bit 0.

†The Track Address of the Selected Drive – Error is only meaningful on a code 150 error. The register contains the address of the cylinder that the head reached on a seek error.

When the RX02 senses the return of power, it will remove Done and begin a sequence to:

1. Move each drive head position mechanism to track 0
2. Clear any active error bits
3. Read sector 1 of track 1, on drive 0
4. Assert Initialize Done in the RXES.

Upon completion of the power up sequence, 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.

**4.3.3.9 RX02 Power Fail** – When the RX02 control senses a loss of power within the RX02, it will unload the head and abort all controller action. The RXAC L line is asserted to indicate to the RX211/RXV21 that subsystem power is gone. The RX211/RXV21 asserts Done and Error and sets the RXAC L bit in the RX2ES.

### 4.3.4 Error Recovery

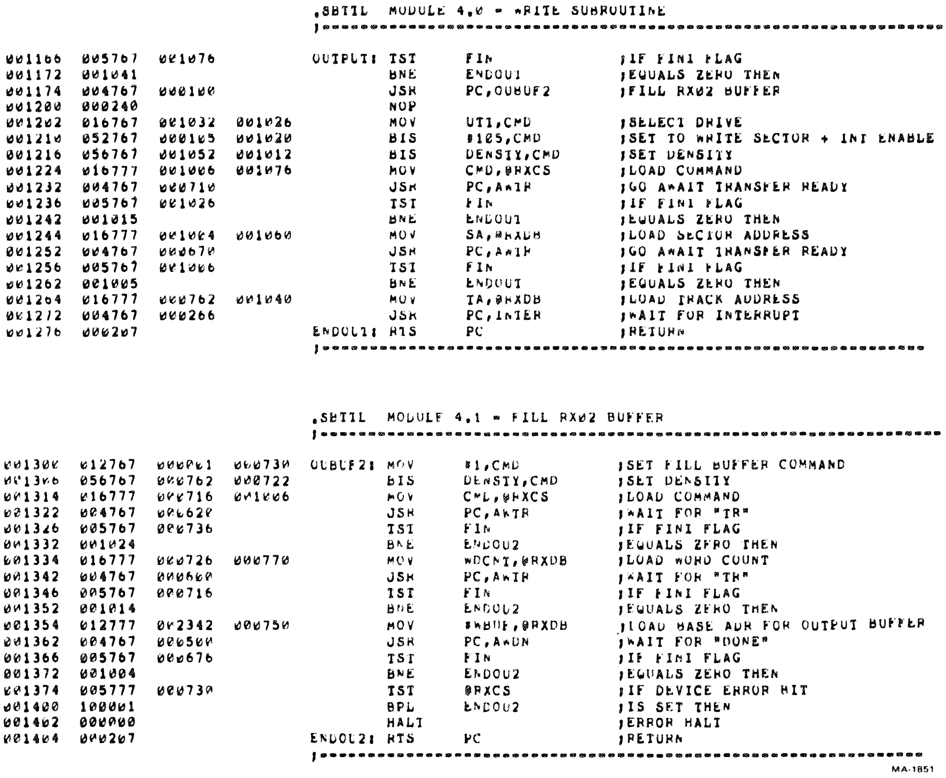
There are two error indications given by the RX211/RXV21 system. The maintenance read status function (Paragraph 4.3.3.6) will assemble the current contents of the RX2ES which can be sampled to determine errors. The read error code function (Paragraph 4.3.3.8) can also be retrieved for explicit error information. The RX211/RXV21 interface register can be interrogated to determine the type of failure that occurred. The error codes and their meaning are listed below.

| **Octal Code** | **Error Code Meaning** |
|---|---|
| 0010 | Drive 0 failed to see home on Initialize. |
| 0020 | Drive 1 failed to see home on Initialize. |
| 0040 | Tried to access a track greater than 76 |
| 0050 | Home was found before desired track was reached. |
| 0070 | Desired sector could not be found after looking at 52 headers (2 revolutions). |
| 0110 | More than 40 $\mu$s and no SEP clock seen |
| 0120 | A preamble could not be found. |
| 0130 | Preamble found but no ID mark found within allowable time span |
| 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. |
| 0220 | R/W electronics failed maintenance mode test. |
| 0230 | Word count overflow |
| 0240 | Density Error |
| 0250 | Wrong key word for set media density command |

### 4.3.5 RX211/RXV21Programming Examples

#### 4.3.5.1 Write/Fill Buffer

Figure 4-31 illustrates a program to write data on a disk by performing write and fill buffer subroutines. Initially, the write subroutine tests to see if there is an error from the last operation. If there is an error, a branch is made and the write subroutine is not performed; otherwise a jump is made to the fill buffer subroutine. (Before data can be written the RX02 sector buffer must be filled.) The Fill Buffer command is set, the density (single or double) is set, and the command is loaded in the RX02/RXCS. After a TR is received, the word count (for either 128 or 256 bytes of data) is loaded in the RX02/RXDB. After another TR is received, the starting address where data will be retrieved from memory is loaded in the RX02/RXDB. The RX02 controller fills the sector buffer with the number of bytes indicated then the RX02 controller sets the Done bit. (If an Error is detected, the Error bit is set in the RXCS and the program halts.) The program returns to the write subroutine, the drive is selected, the write command and interrupt enable are set, the density is set, and the command is loaded in the RX02/RXCS. There is a wait for TR, then the sector address is loaded in the RX02/RXDB; there is another wait for TR and the track address is loaded in the RX02/RXDB. The data loaded in the sector buffer is written by the RX02 controller on the selected drive (disk) at the selected track and sector. While the controller writes the data, the program waits for an interrupt (which signifies the completion of write data) to occur in order to return to the main program.



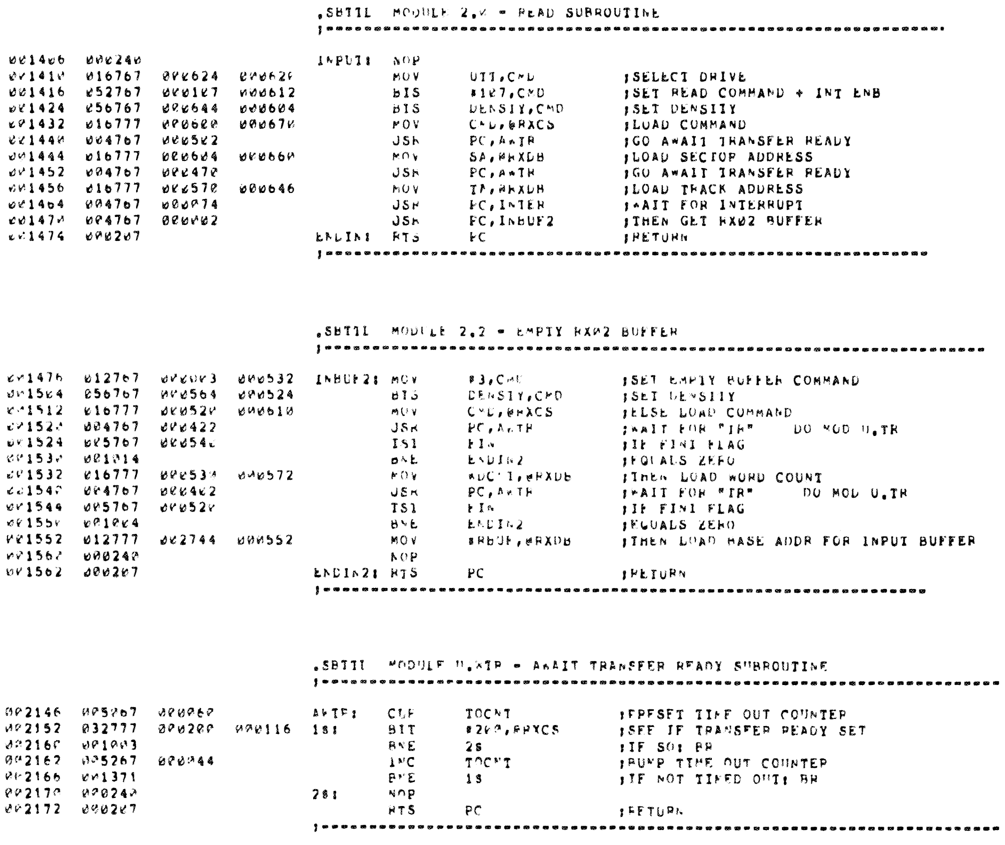
```
                                        .SBTTL  MODULE 4.0 - WRITE SUBROUTINE
                                        ;------------------------------------------------------------

001166  005767  001076                  OUTPUT: TST     FIN             ;IF FINI FLAG
001172  001041                                  BNE     ENDOU1          ;EQUALS ZERO THEN
001174  004767  000100                          JSR     PC,OUBUF2       ;FILL RX02 BUFFER
001200  000240                                  NOP
001202  016767  001032  001026                  MOV     UT1,CMD         ;SELECT DRIVE
001210  052767  000105  001020                  BIS     #105,CMD        ;SET TO WRITE SECTOR + INT ENABLE
001216  056767  001052  001012                  BIS     DENSTY,CMD      ;SET DENSITY
001224  016777  001006  001076                  MOV     CMD,@RXCS       ;LOAD COMMAND
001232  004767  000710                          JSR     PC,AWTR         ;GO AWAIT TRANSFER READY
001236  005767  001026                          TST     FIN             ;IF FINI FLAG
001242  001015                                  BNE     ENDOU1          ;EQUALS ZERO THEN
001244  016777  001004  001060                  MOV     SA,@RXDB        ;LOAD SECTOR ADDRESS
001252  004767  000670                          JSR     PC,AWTR         ;GO AWAIT TRANSFER READY
001256  005767  001006                          TST     FIN             ;IF FINI FLAG
001262  001005                                  BNE     ENDOUT          ;EQUALS ZERO THEN
001264  016777  000762  001040                  MOV     TA,@RXDB        ;LOAD TRACK ADDRESS
001272  004767  000266                          JSR     PC,INTER        ;WAIT FOR INTERRUPT
001276  000207                          ENDOU1: RTS     PC              ;RETURN
                                        ;------------------------------------------------------------


                                        .SBTTL  MODULE 4.1 - FILL RX02 BUFFER
                                        ;------------------------------------------------------------

001300  012767  000001  000730          OUBUF2: MOV     #1,CMD          ;SET FILL BUFFER COMMAND
001306  056767  000762  000722                  BIS     DENSTY,CMD      ;SET DENSITY
001314  016777  000716  001006                  MOV     CMD,@RXCS       ;LOAD COMMAND
001322  004767  000620                          JSR     PC,AWTR         ;WAIT FOR "TR"
001326  005767  000736                          TST     FIN             ;IF FINI FLAG
001332  001024                                  BNE     ENDOU2          ;EQUALS ZERO THEN
001334  016777  000726  000770                  MOV     WDCNT,@RXDB     ;LOAD WORD COUNT
001342  004767  000600                          JSR     PC,AWTR         ;WAIT FOR "TR"
001346  005767  000716                          TST     FIN             ;IF FINI FLAG
001352  001014                                  BNE     ENDOU2          ;EQUALS ZERO THEN
001354  012777  002342  000750                  MOV     #WBUF,@RXDB     ;LOAD BASE ADR FOR OUTPUT BUFFER
001362  004767  000500                          JSR     PC,AWDN         ;WAIT FOR "DONE"
001366  005767  000676                          TST     FIN             ;IF FINI FLAG
001372  001004                                  BNE     ENDOU2          ;EQUALS ZERO THEN
001374  005777  000730                          TST     @RXCS           ;IF DEVICE ERROR BIT
001400  100001                                  BPL     ENDOU2          ;IS SET THEN
001402  000000                                  HALT                    ;ERROR HALT
001404  000207                          ENDOU2: RTS     PC              ;RETURN
                                        ;------------------------------------------------------------
```

MA-1851

Figure 4-31 RX211/RXV21 Write/Fill Buffer Example

### 4.3.5.2 Read/Empty Buffer

Figure 4-32 illustrates a program to read data from the disk by performing read and empty buffer subroutines. The drive to be read is selected, the read command and interrupt enable are set, the density is set, and the command is loaded in the RX02/RXCS. There is a wait for TR and then the sector address is loaded in the RX02/RXDB; there is another wait for TR, and the track address is loaded in the RX02/RXDB. While the RX02 controller reads data from the selected location on the selected disk into the RX02 sector buffer, the program waits for an interrupt to occur and then there is a jump to the empty buffer subroutine. The empty buffer command is set, the density is set, and the command is loaded into the RX02/RXCS. After a TR is received, the word count is loaded into the RX02/RXDB; there is another wait for TR and the address in memory where the data is to be stored is loaded into the RX02/RXDB. The data is emptied from the sector buffer by the RX02 controller, and when the buffer is emptied, there is a return to the main program.



```
                                        .SBTTL  MODULE 2.0 - READ SUBROUTINE
                                        ;--------------------------------------------------------------------

001406  000240                          INPUT:  NOP
001410  016767  000624  000620                  MOV     UTT,CMD            ;SELECT DRIVE
001416  052767  000107  000612                  BIS     #107,CMD           ;SET READ COMMAND + INT ENB
001424  056767  000644  000604                  BIS     DENSITY,CMD        ;SET DENSITY
001432  016777  000600  000670                  MOV     CMD,@RXCS          ;LOAD COMMAND
001440  004767  000502                          JSR     PC,AWTR            ;GO AWAIT TRANSFER READY
001444  016777  000604  000660                  MOV     SA,@RXDB           ;LOAD SECTOR ADDRESS
001452  004767  000470                          JSR     PC,AWTR            ;GO AWAIT TRANSFER READY
001456  016777  000570  000646                  MOV     TA,@RXDB           ;LOAD TRACK ADDRESS
001464  004767  000074                          JSR     PC,INTER           ;WAIT FOR INTERRUPT
001470  004767  000002                          JSR     PC,INBUF2          ;THEN GET RX02 BUFFER
001474  000207                          ENDIN:  RTS     PC                 ;RETURN
                                        ;--------------------------------------------------------------------


                                        .SBTTL  MODULE 2.2 - EMPTY RX02 BUFFER
                                        ;--------------------------------------------------------------------

001476  012767  000003  000532          INBUF2: MOV     #3,CMD             ;SET EMPTY BUFFER COMMAND
001504  056767  000564  000524                  BIS     DENSITY,CMD        ;SET DENSITY
001512  016777  000520  000510                  MOV     CMD,@RXCS          ;ELSE LOAD COMMAND
001520  004767  000422                          JSR     PC,AWTR            ;WAIT FOR "TR"     DO MOD U.TR
001524  005767  000540                          TST     FIN                ;IF FINI FLAG
001530  001014                                  BNE     ENDIN2             ;EQUALS ZERO
001532  016777  000534  000572                  MOV     WDC'1,@RXDB        ;THEN LOAD WORD COUNT
001540  004767  000402                          JSR     PC,AWTR            ;WAIT FOR "TR"     DO MOD U.TR
001544  005767  000520                          TST     FIN                ;IF FINI FLAG
001550  001004                                  BNE     ENDIN2             ;EQUALS ZERO
001552  012777  002744  000552                  MOV     #RBUF,@RXDB        ;THEN LOAD BASE ADDR FOR INPUT BUFFER
001560  000240                                  NOP
001562  000207                          ENDIN2: RTS     PC                 ;RETURN
                                        ;--------------------------------------------------------------------


                                        .SBTTL  MODULE U.WTR - AWAIT TRANSFER READY SUBROUTINE
                                        ;--------------------------------------------------------------------

002146  005067  000062                  AWTR:   CLR     TOCNT              ;PRESET TIME OUT COUNTER
002152  032777  000200  000116          1$:     BIT     #200,@RXCS         ;SEE IF TRANSFER READY SET
002160  001003                                  BNE     2$                 ;IF SO: BR
002162  005267  000044                          INC     TOCNT              ;BUMP TIME OUT COUNTER
002166  001371                                  BNE     1$                 ;IF NOT TIMED OUT: BR
002170  000240                          2$:     NOP
002172  000207                                  RTS     PC                 ;RETURN
                                        ;--------------------------------------------------------------------
```

MA-1852

Figure 4-32 RX211/RXV21 Read/Empty Buffer Example

RX02 FLOPPY DISK SYSTEM
USER'S GUIDE
EK-RX02-UG-001

# Reader's Comments

**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 or errors have you found in the manual? ______________________________

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

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

□ Please send me the current copy of the *Technical Documentation Catalog*, which contains information on the remainder of DIGITAL's technical documentation.

Name ______________ Street ______________
Title ______________ City ______________
Company ______________ State/Country ______________
Department ______________ Zip ______________

Additional copies of this document are available from:

Digital Equipment Corporation
444 Whitney Street
Northboro, Ma 01532
Attention: Communications Services (NR2/M15)
Customer Services Section

Order No. EK-RX02-UG-001

— — — — — — — — — — — Fold Here — — — — — — — — — — — —

— — — — — — — — Do Not Tear - Fold Here and Staple — — — — — — — —

Digital Equipment Corporation
Technical Documentation Department
Maynard, Massachusetts 01754