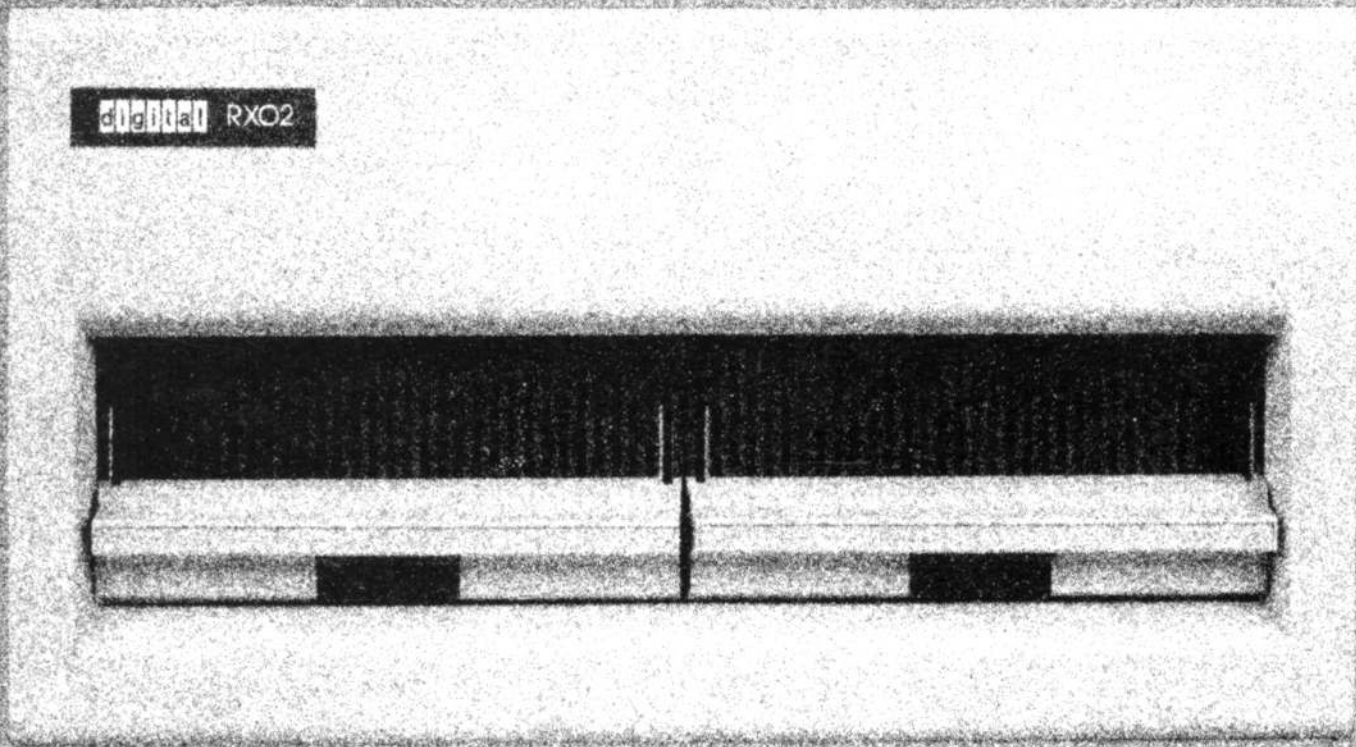
digital

RX02
FLOPPY DISK SYSTEM
TECHNICAL MANUAL



EK-0RX02-TM-001

RX02 Floppy Disk System Technical Manual

digital equipment corporation • maynard, massachusetts

1st Edition, October 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.

CONTENTS

CONTENTS (Cont)

CONTENTS (Cont)

CONTENTS (Cont)

CONTENTS (Cont)

CONTENTS (Cont)

CONTENTS (Cont)

FIGURES

FIGURES (Cont)

FIGURES (Cont)

TABLES

PREFACE

The 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, programming and maintenance 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 one or two flexible disk drives, a single read/write electronics module, a microprogrammed controller module, and a power supply, enclosed in a rack-mountable, 10-1/2 inch, self-cooled chassis. A cable is included for connection to either a PDP-8 interface module, 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. 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 128 12-bit words (PDP-8) can be stored and retrieved. The RX02 interfaces with IBM-compatible devices when single density data recording is used. If two drives are used, the recording density can be different for each drive.

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.

1.2.2 Microprogrammed Controller

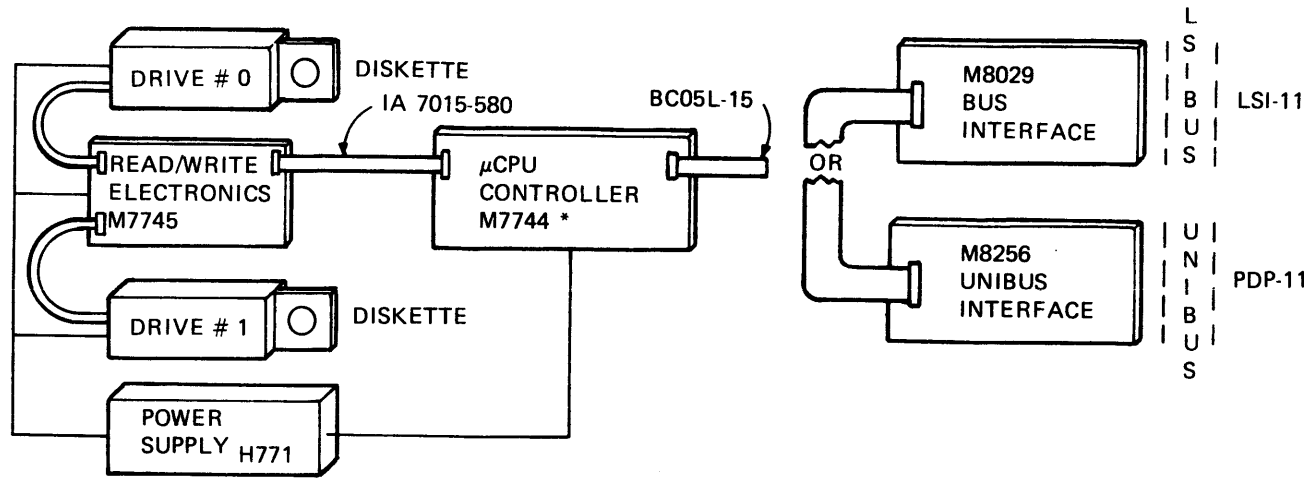
The M7744 microprogrammed controller module is located in the RX02 cabinet as shown in Figure 1-4. The M7744 is hinged on the left side and lifts up for access to the M7745 read/write electronics module.

1.2.3 Read/Write Electronics

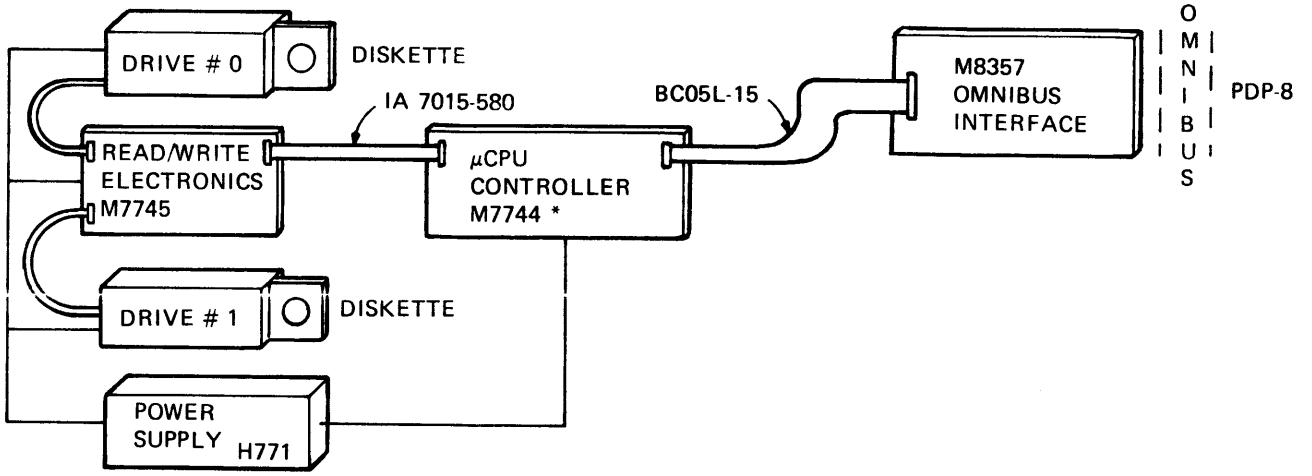
The M7745 read/write electronics module is located in the RX02 cabinet as shown in Figure 1-4.

1.2.4 Electromechanical Drive

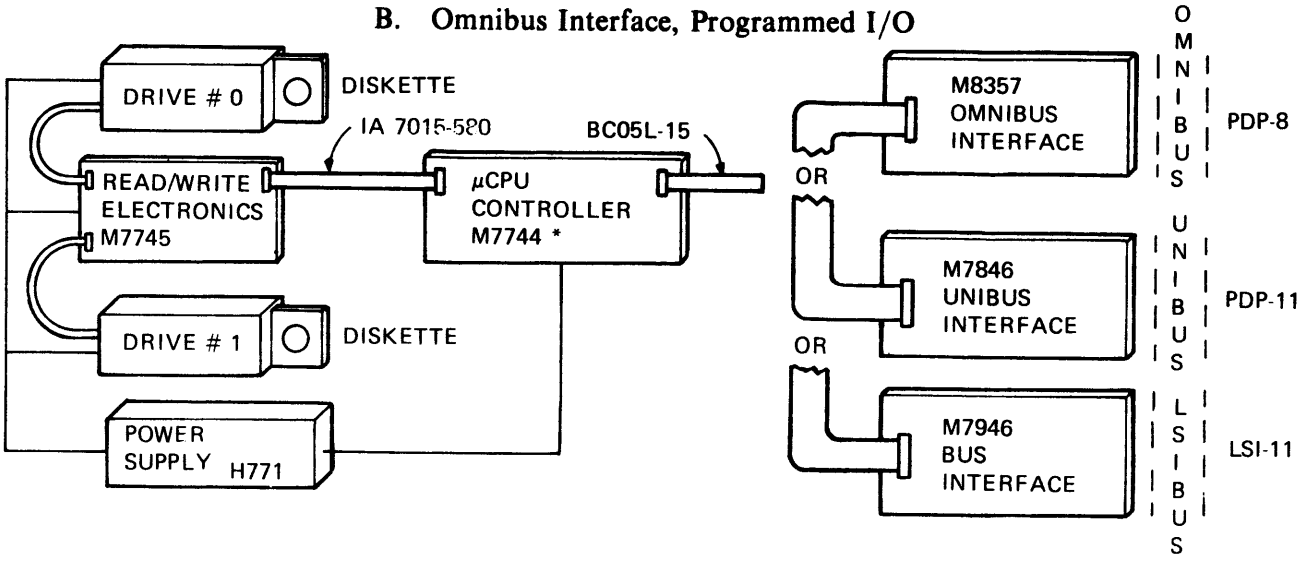
A maximum of two drives can be attached to the read/write electronics. The electromechanical drives are mounted side by side under the read/write electronics board (M7745). Figure 1-5 is an underside view of the drive showing the drive motor connected to the spindle by a belt. (This belt and the drive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



A. Bus Interface with DMA



B. Omnibus Interface, Programmed I/O



C. Bus Interface, Programmed I/O (RX01 Compatible)

Figure 1-1 Floppy Disk Configuration

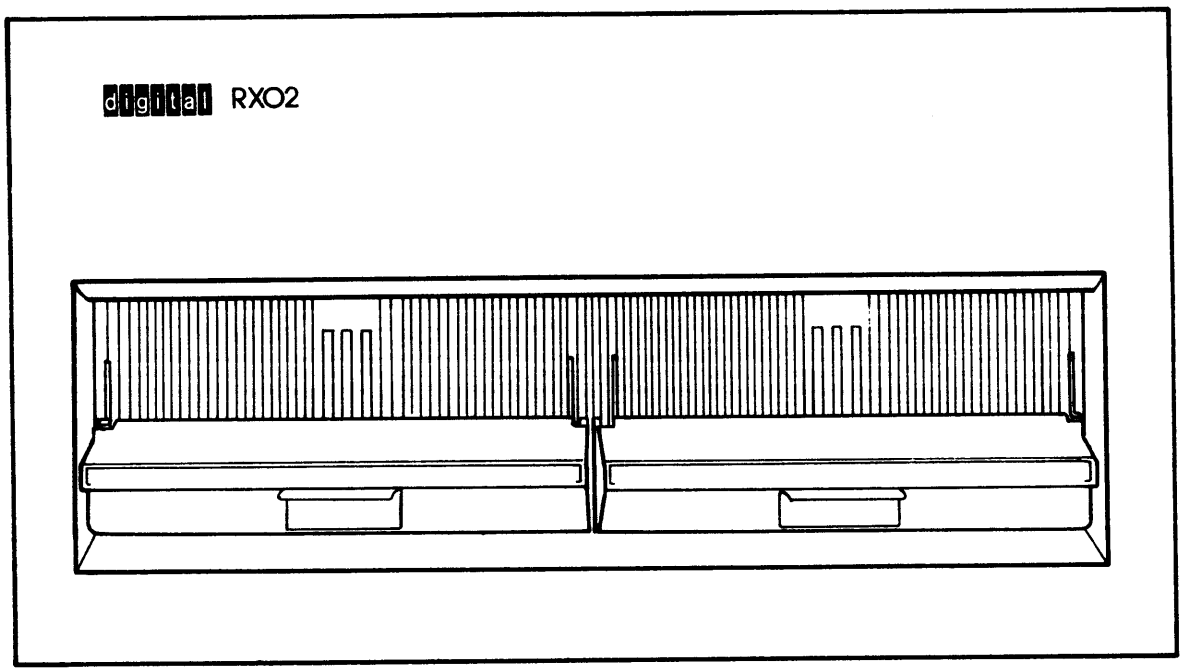


Figure 1-2 Front View of the Floppy Disk System

1.2.5 Power Supply

The H771 power supply is mounted at the rear of the 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 100–128 Vac	3.5 A circuit breaker; H771-C
184–240 Vac 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.

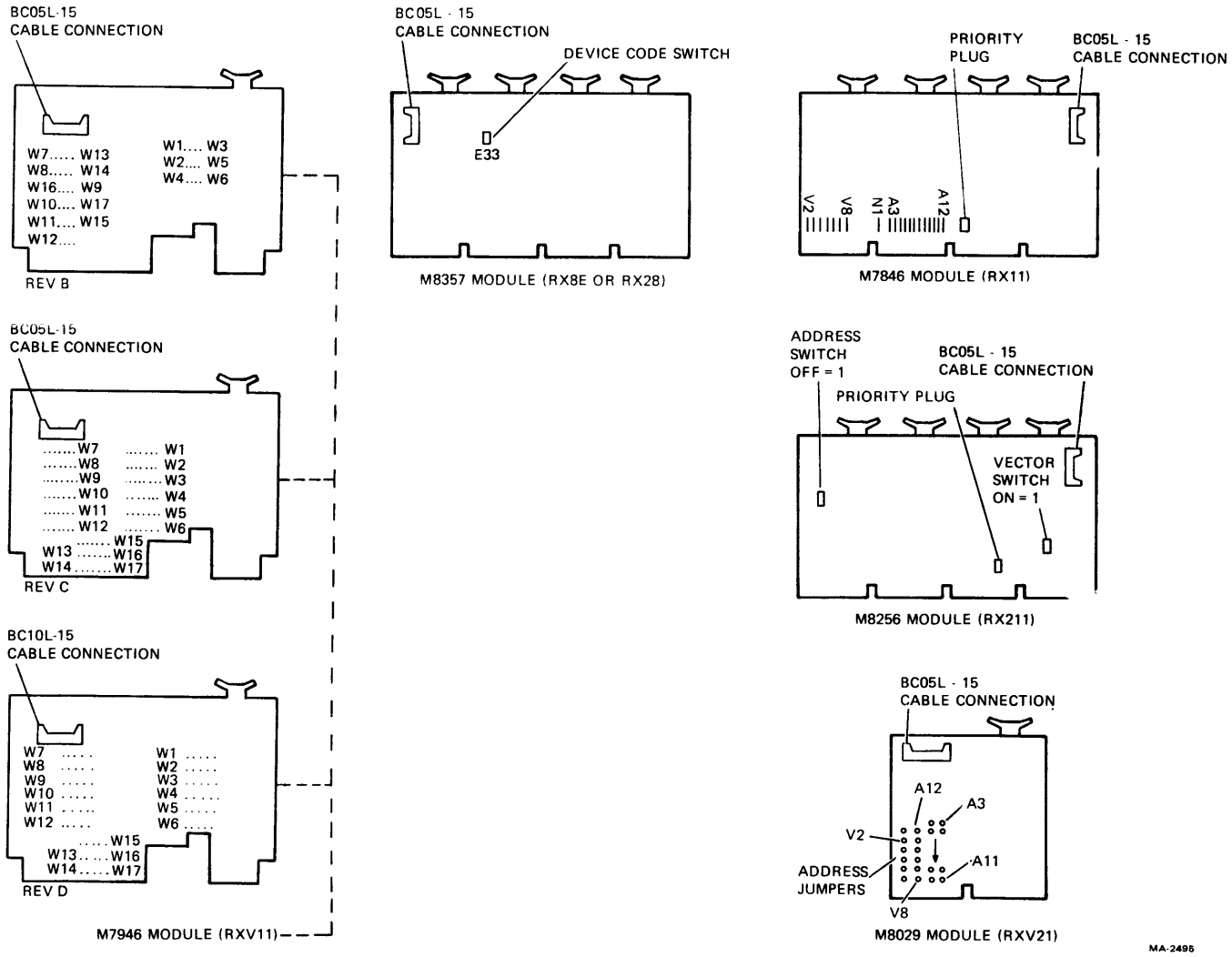


Figure 1-3 Interface Modules

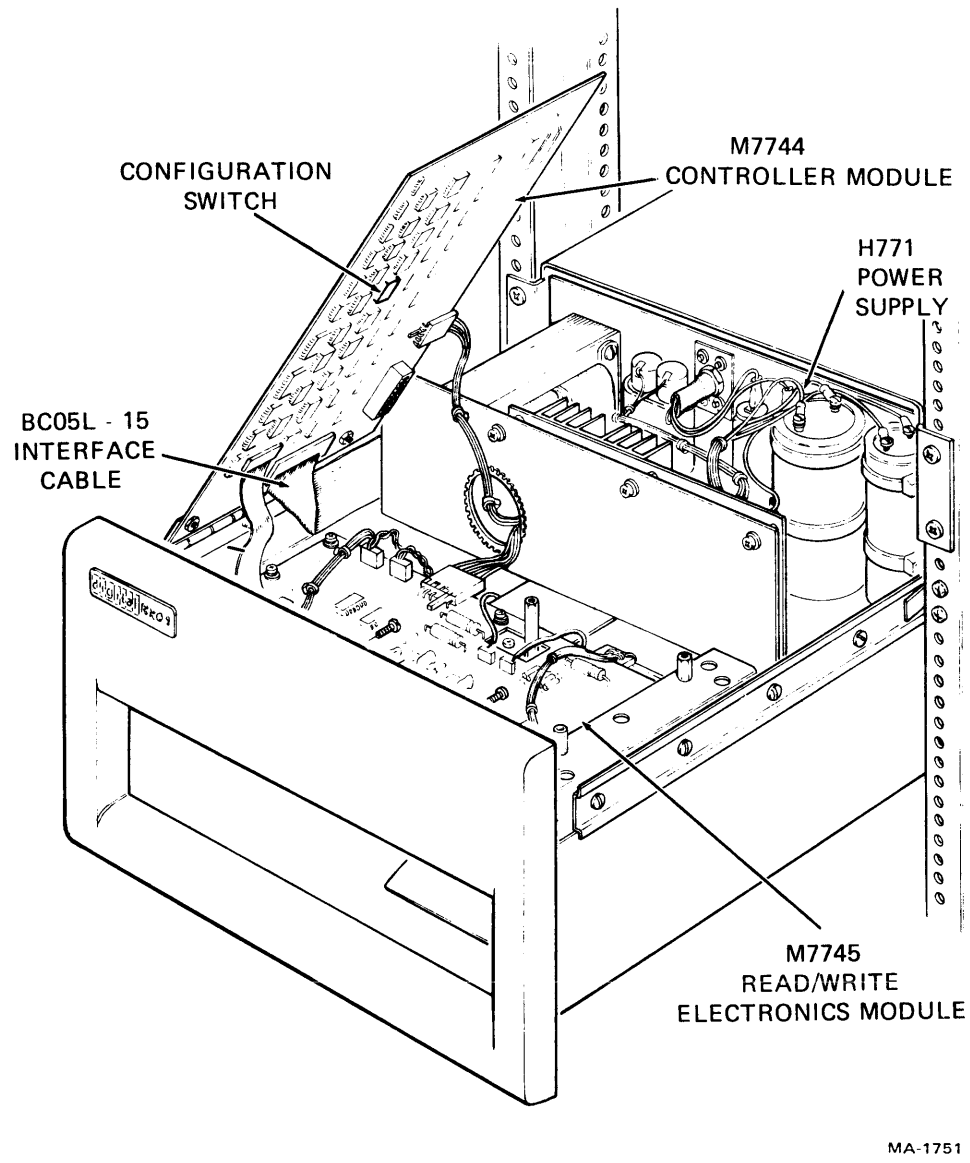


Figure 1-4 Top View of RX02

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.

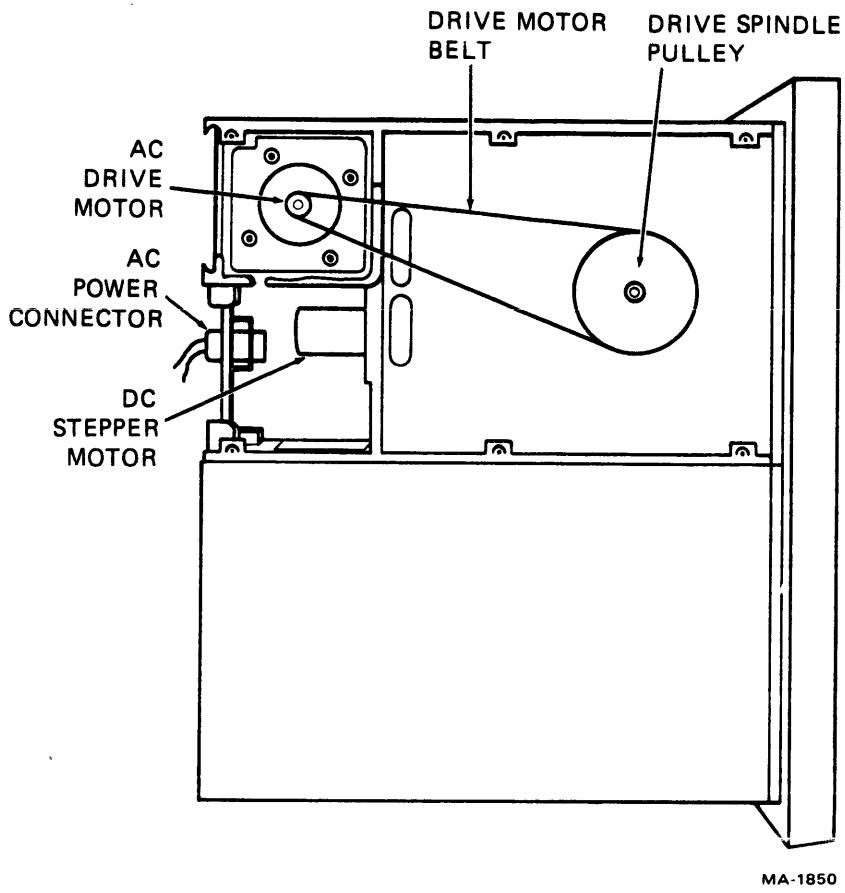


Figure 1-5 Underside View of Drive

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) 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:

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

Environmental Characteristics

Temperature	
RX02, operating	15° to 32° C (59° to 90° F) ambient; 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 2° C (36° F) minimum dew point 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 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.

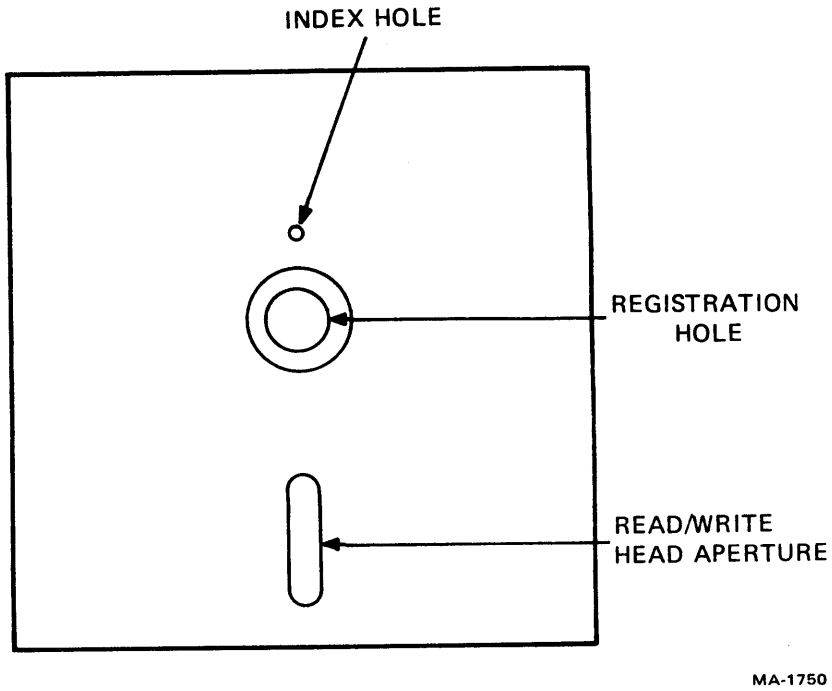


Figure 1-6 Diskette Media

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 "IF" or fundamental frequency bit stream.

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.

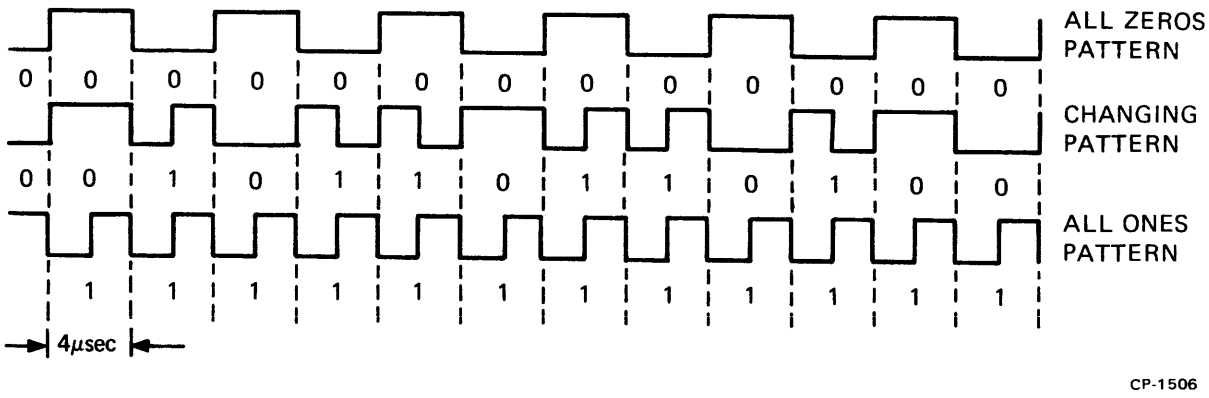


Figure 1-7 Flux Reversal Patterns for FM

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.

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).

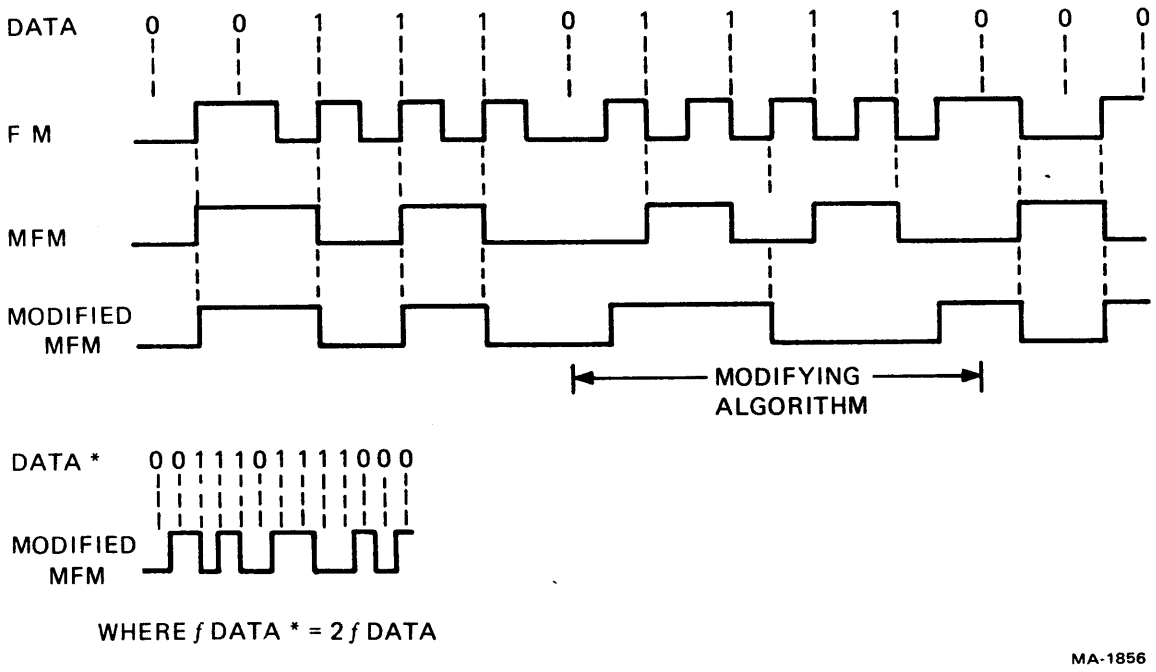


Figure 1-8 FM Versus MFM Encoding

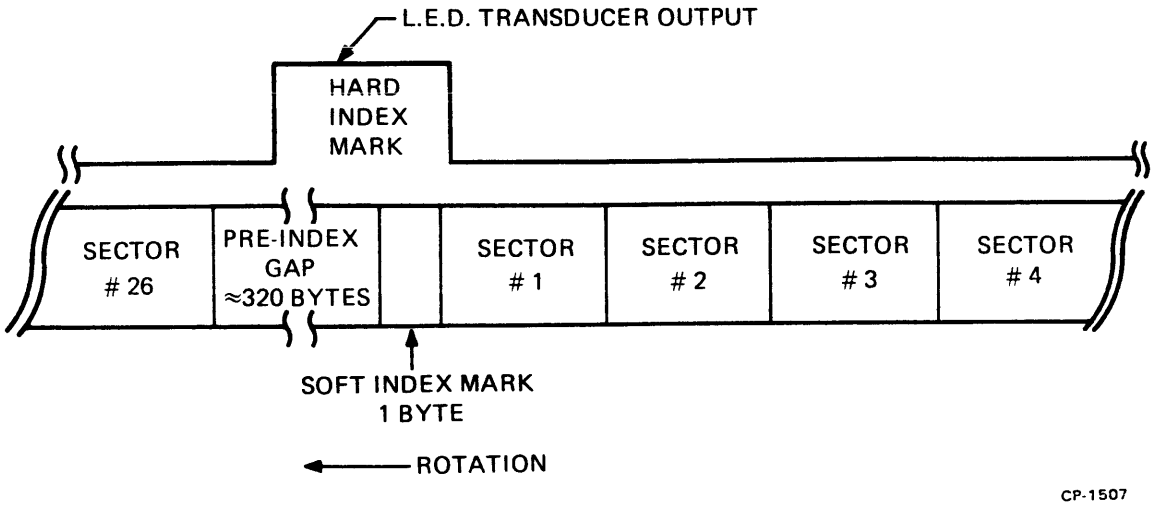


Figure 1-9 Track Format (Each Track)

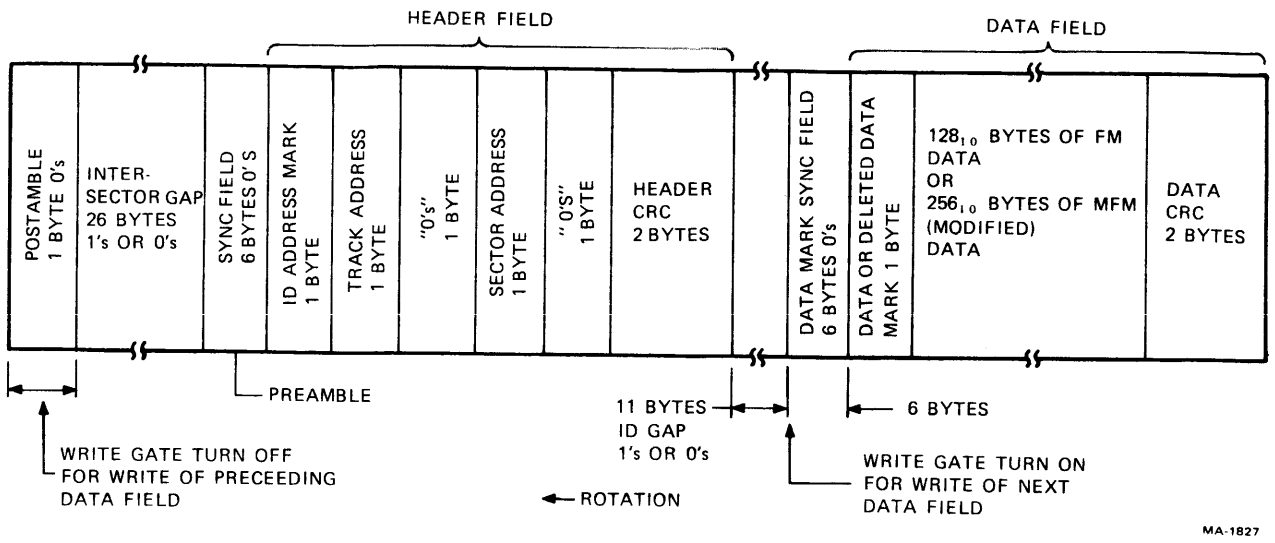


Figure 1-10 Sector Format (Each Sector)

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

2.1 SITE PREPARATION

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.

2.1.1 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.

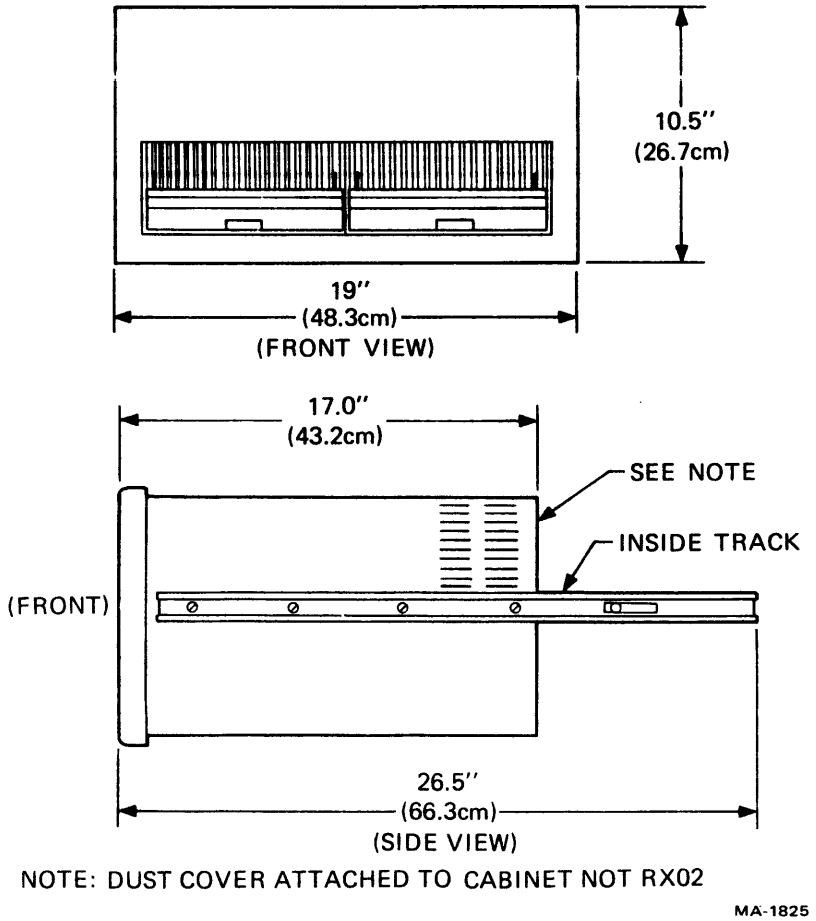


Figure 2-1 RX02 Outline Dimensions

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.

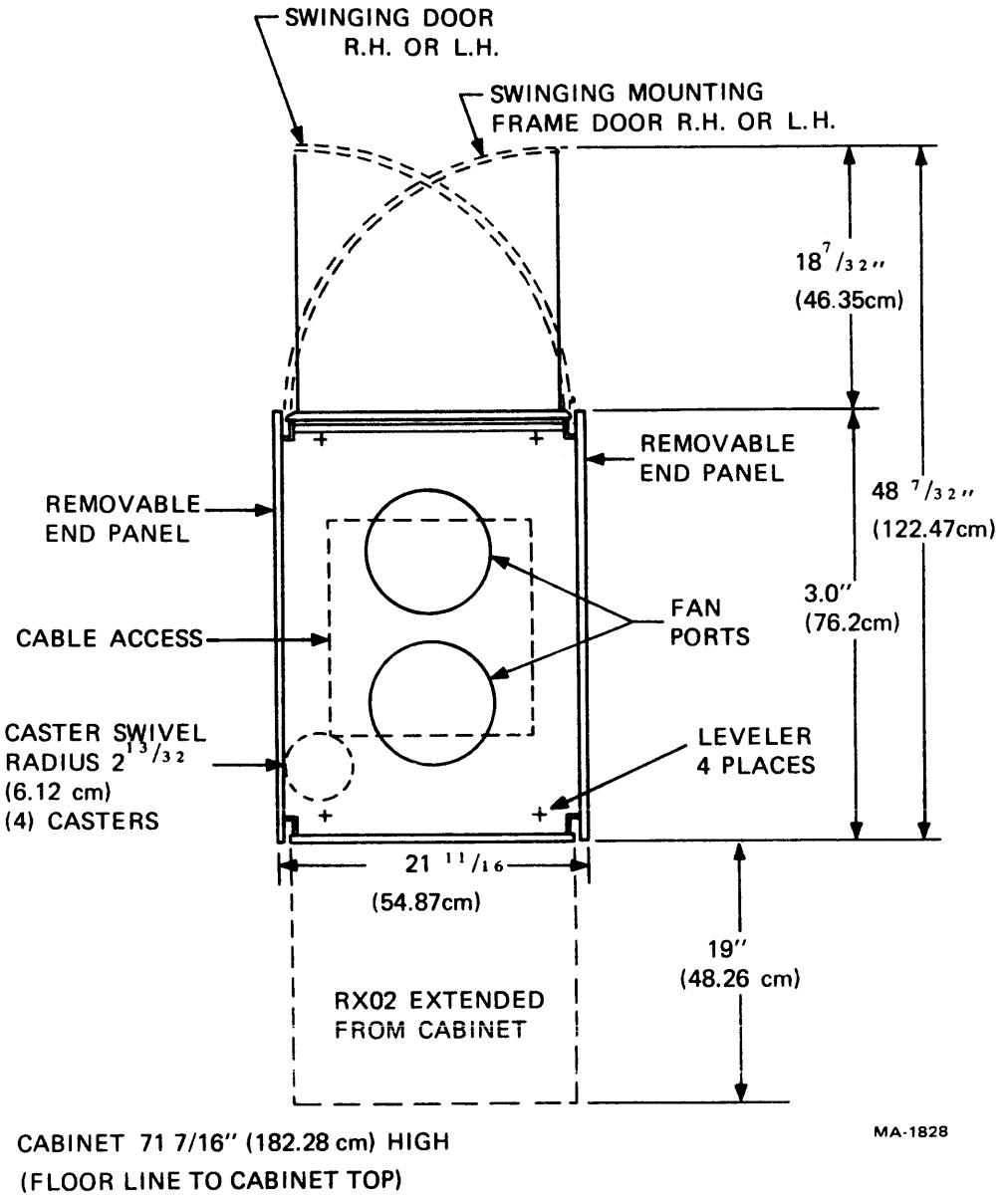


Figure 2-2 Cabinet Layout Dimensions

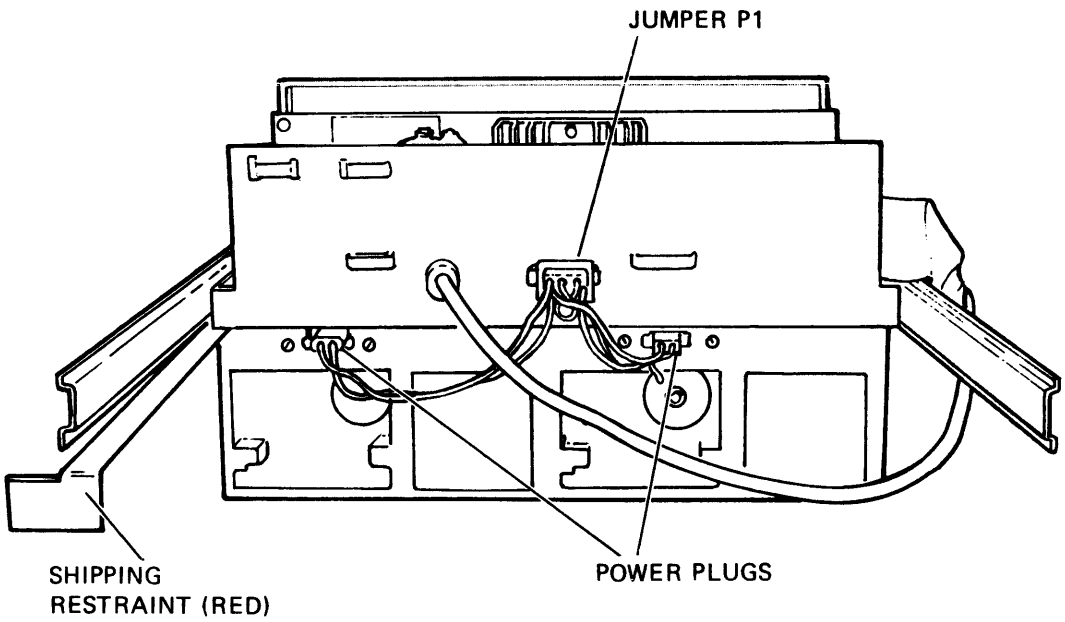
2.1.2 Cabling

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, 12-12301-01
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.)

NOTE

For single drive configurations, the drive will be identified as drive 0 and will be mounted as the left drive. For dual drive configurations, the left drive will be identified as drive 0 and the right drive will be identified as drive 1.

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 radio broadcasts, 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.

Table 2-1 RX02 Configurations

Computer	System Designation	μCPU Controller	Interface Module	RX02 Model No.	Drive
	RX8E	M7744	M8357	RX02-AA	Single/115 V, 60 Hz
				RX02-AC	Single/115 V, 50 Hz
				RX02-AD	Single/230 V, 50 Hz
				RX02-BA	Dual/115 V, 60 Hz
				RX02-BC	Dual/115 V, 50 Hz
				RX02-BD	Dual/230 V, 50 Hz
PDP-8	RX28E	M7744	M8357	RX02-AA	Single/115 V, 60 Hz
				RX02-AC	Single/115 V, 50 Hz
				RX02-AD	Single/230 V, 50 Hz
				RX02-BA	Dual/115 V, 60 Hz
				RX02-BC	Dual/115 V, 50 Hz
				RX02-BD	Dual/230 V, 50 Hz
	RX11	M7744	M7846	RX02-AA	Single/115 V, 60 Hz
				RX02-AC	Single/115 V, 50 Hz
				RX02-AD	Single/230 V, 50 Hz
				RX02-BA	Dual/115 V, 60 Hz
				RX02-BD	Dual/230 V, 50 Hz
PDP-11	RX211	M7744	M8256	RX02-AA	Single/115 V, 60 Hz
				RX02-AC	Single/115 V, 50 Hz
				RX02-AD	Single/230 V, 50 Hz
				RX02-BA	Dual/115 V, 60 Hz
				RX02-BC	Dual/115 V, 50 Hz
				RX02-BD	Dual/230 V, 50 Hz
	RXV11	M7744	M7946	RX02-AA	Single/115 V, 60 Hz
				RX02-AC	Single/115 V, 50 Hz
				RX02-AD	Single/230 V, 50 Hz
				RX02-BA	Dual/115 V, 60 Hz
				RX02-BC	Dual/115 V, 50 Hz
				RX02-BD	Dual/230 V, 50 Hz
LSI-11	RXV21	M7744	M8029	RX02-AA	Single/115 V, 60 Hz
				RX02-AC	Single/115 V, 50 Hz
				RX02-AD	Single/230 V, 50 Hz
				RX02-BA	Dual/115 V, 60 Hz
				RX02-BC	Dual/115 V, 50 Hz
				RX02-BD	Dual/230 V, 50 Hz

Table 2-2 Controller Configuration Switch Positions

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

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).

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).

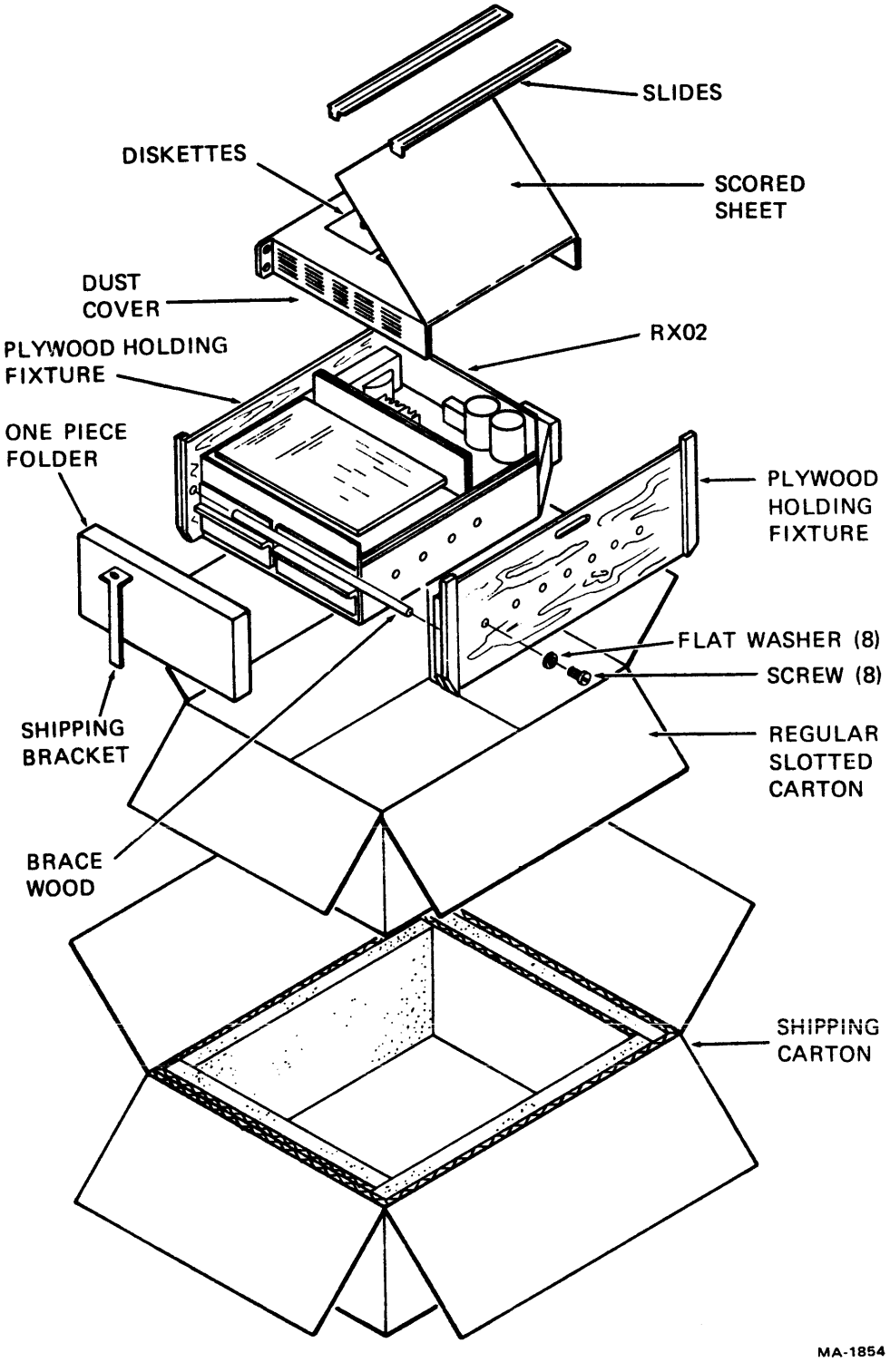


Figure 2-4 RX02 Unpacking

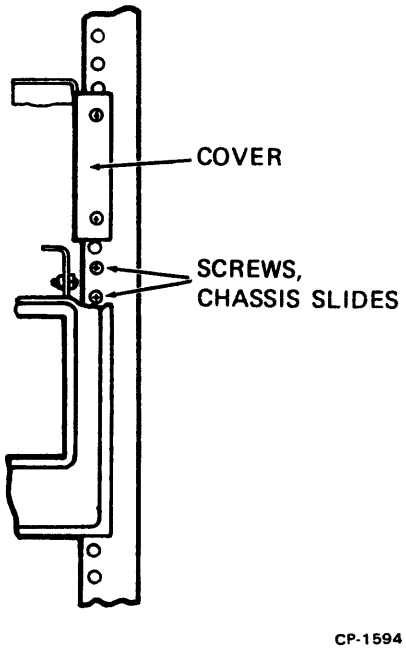


Figure 2-5 RX02 Cabinet Mounting Information

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 - Removed	SW10 OFF	V2 - Installed	SW1 ON
BR6 - 54-08780	A11 - Removed	SW9 OFF	V3 - Removed	SW2 OFF
BR5 - 54-08778*	A10 - Removed	SW8 OFF	V4 - Installed	SW3 ON
BR4 - 57-08776	A9 - Removed	SW7 OFF	V5 - Installed	SW4 ON
	A8 - Installed	SW6 ON	V6 - Removed	SW5 OFF
	A7 - Installed	SW5 ON	V7 - Installed	SW6 ON
	A6 - Removed	SW4 OFF	V8 - Removed	SW7 OFF
	A5 - Removed	SW3 OFF		
	A4 - Removed	SW2 OFF		
	A3 - Removed	SW1 OFF		

LSI-11 (M8029)

	Register Address* (17717X)	Vector Address (264_8)
	A12 - Installed	V2 - Installed
	A11 - Installed	V3 - Removed
	A10 - Installed	V4 - Installed
	A9 - Installed	V5 - Installed
	A8 - Removed	V6 - Removed
	A7 - Removed	V7 - Installed
	A6 - Installed	V8 - Removed
	A5 - Installed	
	A4 - Installed	
	A3 - Installed	
	*Standard	

Table 2-3 Interface Code/Jumper Configuration (Cont)

LSI-11 (M7946)			
Vector Address		Register Address	
264_8	270_8	177170	177150
W6 - Removed	W6 - Removed	W17 - Removed	W17 - Removed
W5 - Installed	W5 - Installed	W16 - Removed	W16 - Removed
W4 - Removed	W4 - Removed	W14 - Removed	W14 - Removed
W3 - Removed	W3 - Removed	W13 - Installed	W13 - Installed
W2 - Installed	W2 - Removed	W11 - Removed	W11 - Removed
W1 - Removed	W1 - Installed	W10 - Removed	W10 - Removed
		W9 - Removed	W9 - Installed
		W8 - Removed	W8 - Removed
		W7 - Installed	W7 - Installed

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 the problem to DIGITAL Field Service.

Table 3-1 Operator Troubleshooting Guide

Problem	Probable Cause	Correction
No power (drive inoperative)	a. Power cord disconnected b. Blown fuse c. Circuit breaker open	a. Connect power cord b. Replace fuse c. Close circuit breaker
Drive not ready	a. Drive door open b. Diskette improperly installed	a. Close door b. Properly seat diskette
Error in recording	a. Diskette wear b. Diskette mounting hole c. Mismatch in recording density on a diskette	a. If worn, replace b. If the hole is not concentric, replace diskette 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 first, 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.)

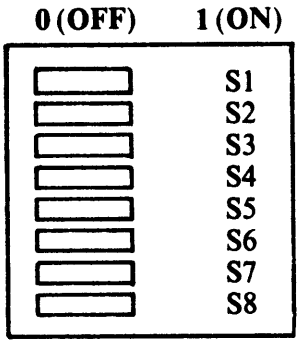
4.1.1 Device Codes

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.

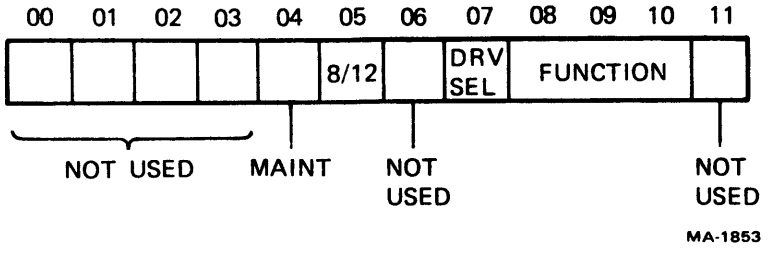


Figure 4-1 LCD Word Format (RX8E)

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.

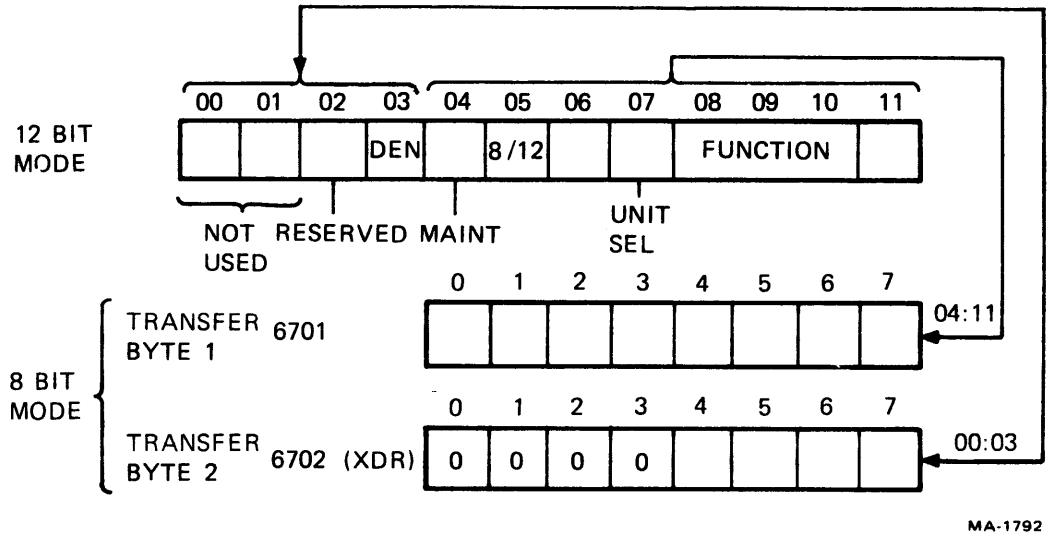


Figure 4-2 Command Word Format (RX28)

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).

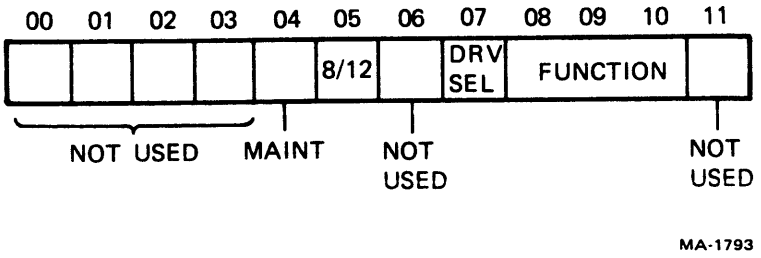


Figure 4-3 Command Register Format (RX8E)

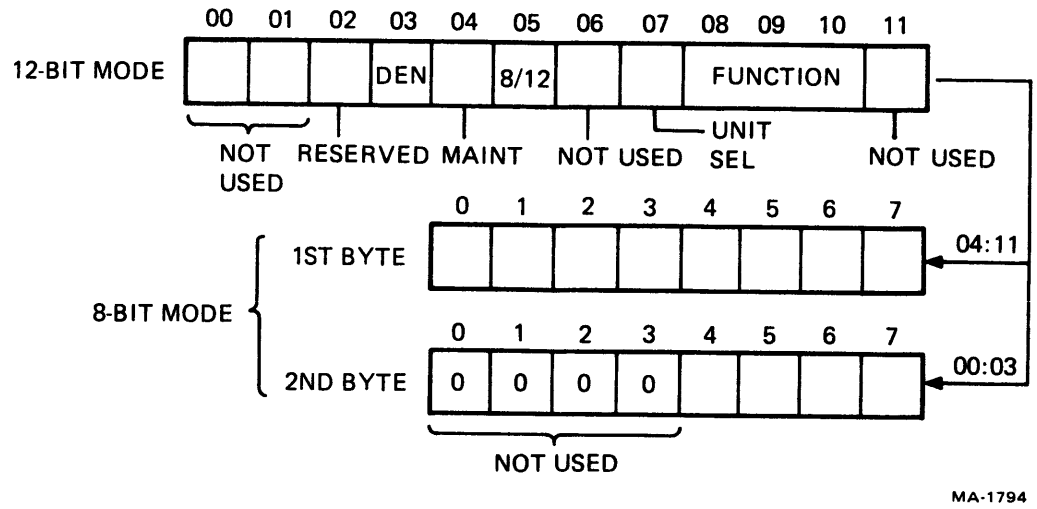


Figure 4-4 Command Register Format (RX28F)

The function codes (bits 8, 9, 10) are suummarized 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 Errror 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.

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.

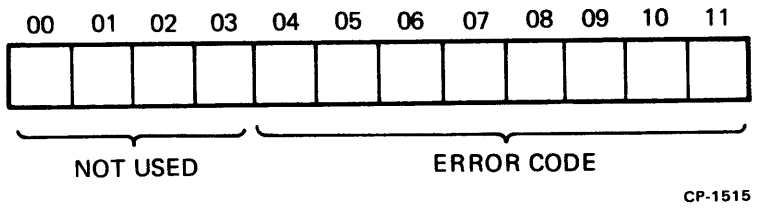


Figure 4-5 Error Code Register Format (RX8E/RX28A)

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.

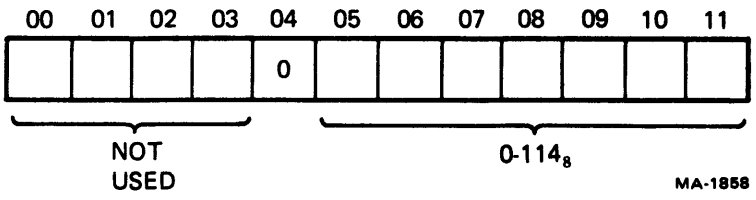


Figure 4-6 RX2TA Format (RX8E/RX28)

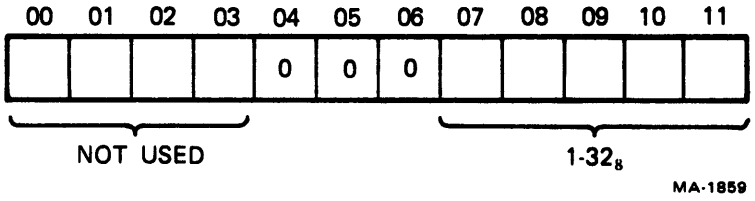


Figure 4-7 RX2SA Format (RX8E/RX28)

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.

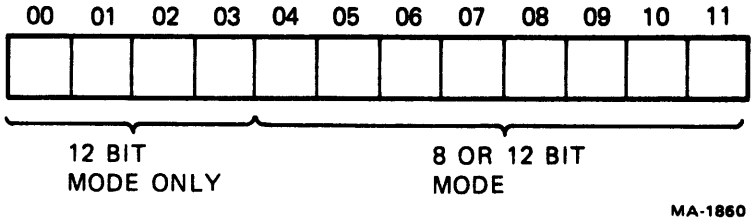


Figure 4-8 RX2DB Format (RX8E/RX28)

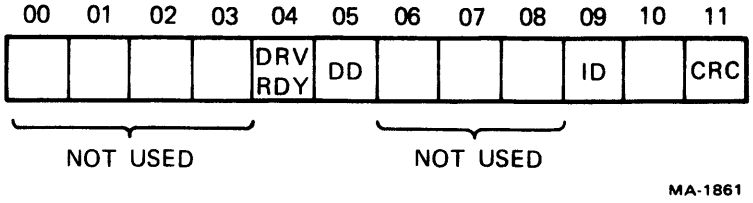


Figure 4-9 RXES Format (RX8E)

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.

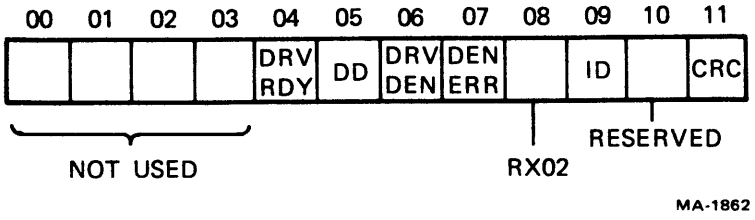


Figure 4-10 RX2ES Format (RX28)

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.
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 thee 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.

Octal	8	AC 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.

```
                /PROGRAMMING EXAMPLES FOR THE RX8/RX01 FLEXIBLE DISKETTE
                /
                /THE FOLLOWING ARE RX01 IOT CODE DEFINITIONS
                /
                /THE STANDARD IOT DEVICE CODE IS 670-
                /
        6701    LCD=6701                /IOT TO LOAD THE COMMAND, (AC) IS THE COMMAND
        6702    XDR=6702                /IOT TO LOAD OR READ THE TRANSFER REGISTER
        6703    STR=6703                /IOT TO SKIP ON A TRANSFER REQUEST FLAG
        6704    SER=6704                /IOT TO SKIP ON AN ERROR FLAG
        6705    SDN=6705                /IOT TO SKIP ON THE DONE FLAG
        6706    INTR=6706               / (AC) = 0 INTERRUPT ENABLE OFF/ (AC) = 1 MEANS ON
        6707    INIT=6707               /IOT TO INITIALIAE THE RX8/RX01 SUBSYSTEM
                /
                /THE FOLLOWING IS A PROGRAMMING EXAMPLE OF THE PROTOCOL REQUIRED
                /
                /TO WRITE, WRITE DELETED DATA, OR READ AT SECTOR "S" (THE CONTENTS OF PROGRAM
                /
                /LOCATION SECTOR) OF TRACK "T" (THE CONTENTS OF PROGRAM LOCATION TRACK)
                /
                /IN 8 OR 12 BIT MODE
                /
0220    1254    START,  TAD KM10                        / -12
0221    3255            DCA PTRY                        /PARITY RETRY COUNTER
0222    1254            TAD KM10
0223    3256            DCA CTRY                        /CRC RETRY COUNTER
0224    1254            TAD KM10
0225    3257            DCA STRY                        /SEEK RETRY COUNTER
                /
                /WRITE, WRITE DELETED DATA, OR READ
                /
0226    1260    RETRY,  TAD MODE                        /0 IF 12-BIT, 100 IF 8-BIT
0227    1261            TAD COMMAND                     / 4 IF WRITE, 14 IF WRITE DELETED
                                                        /DATA, OR 6 IF READ
0230    1262            TAD UNIT                        / 0 IF UNIT 0, 20 IF UNIT 1
0231    6701            LCD                             /IOT 67X1 TO LOAD THE COMMAND
                /
                /WAIT FOR THE TRANSFER REQUEST FLAG THEN TRANSFER THE SECTOR ADDRESS
                /
0232    6703            STR                             /IOT 67X3 TO
0233    5232            JMP .-1                         /WAIT FOR TRANSFER REQUEST FLAG
0234    1263            TAD SECTOR                      / 1 TO 32(OCTAL)
0235    6702            XDR                             /IOT TO LOAD SECTOR
0236    7200            CLA                             /CLA BECAUSE IOT XDR DOESN'T
                /
                /WAIT FOR THE TRANSFER REQUEST FLAG THEN TRANSFER THE TRACK ADDRESS
                /
0237    6703            STR                             /IOT 67X3 TO
0240    5237            JMP .-1                         /WAIT FOR TRANSFER REQUEST
0241    1264            TAD TRACK                       / 0 TO 114(OCTAL)
0242    6702            XDR                             /IOT TO LOAD TRACK
0243    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
                /
0244    6705            SDN                             /IOT 67X5 TO
0245    5244            JMP .-1                         /WAIT FOR DONE FLAG
0246    6704            SER                             /IOT 67X4 SAMPLES ERROR FLAG
0247    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
                /
0250    6702            XDR                             /GET CONTENTS OF TR (ERROR STATUS)
0251    3265            DCA ASTATUS                     /AND SAVE
0252    7305            CLL CLA IAC RAL                 / 2
0253    0265            AND ASTATUS                     /TEST FOR PARITY ERROR
0254    7650            SNA CLA                         /SKIP IF PARITY ERROR
0255    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
                /
0256    2255            ISZ PTRY
0257    5206            JMP RETRY                       /RETRY THE COMMAND
0260    7402            HLT                             /HARD PARITY ERROR
                /
                /THE ERROR FLAG IS SET BUT THE ERROR IS NOT A PARITY ERROR
                /
                /TEST FOR A CRC ERROR
                /
0261    7301    TCRC,   CLL CLA IAC                     / 1
0262    0265            AND ASTATUS                     /TEST FOR A CRC ERROR
0263    7650            SNA CLA                         /SKIP IF A CRC ERROR
0264    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 RETRY COUNTER PROGRAM LOCATION " CTRY "
                /
                /AND RETRY THE COMMAND UNTIL THE CRC ERROR RECOVERS
                /
                /OR UNTIL THE CTRY COUNTER OVERFLOWS TO 0
                /
0245  2256          ISZ CTRY
0246  5206          JMP RETRY                                 /RETRY THE COMMAND
0247  7402          HLT                                       /HARD CRC ERROR
                /
                /THE ERROR FLAG IS SET
                /
                /THE ERROR IS (NOT) A PARITY ERROR AND IS (NOT) A CRC ERROR
                /
                /THEREFORE IS MUST BE A SEEK ERROR
                /
                / (CONTENTS OF THE TRANSFER REGISTER BITS 10, AND 11 = 0)
                /
0250  6707    SEEK,   INIT                                    /IOT 67X7 TO INITIALIAE
                /
                /INCREMENT AND TEST THE SEEK ERROR RETRY COUNTER PROGRAM LOCATION " STRY "
                /
                /AND RETRY THE COMMAND UNTIL THE SEEK ERROR RECOVERS
                /
                /OR UNTIL THE CTRY COUNTER OVERFLOWS TO 0
                /
0251  2257          ISZ STRY
0252  5206          JMP RETRY                                 /RETRY THE COMMAND
0253  7402          HLT                                       /HARD SEEK ERROR

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

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

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.

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

Figure 4-12 RX8E Empty Buffer Example

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.

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 cpack.

```
                        /THE FOLLOWING IS A PROGRAMMING EXAMPLE OF PROTOCOL REQUIRED TO
                        /
                        /FILL THE SECTOR BUFFER WITH 64 12-BIT WORDS (12 BIT MODE), OR
                        /
                        /FILL THE SECTOR BUFFER WITH 128 8-BIT BYTES (8 BIT MODE)
                        /
          0010          A10=10
                        /
0266  1254   FENTRY,    TAD KM10                        / 8 TRYS TO FILL THE SECTOR BUFFER
0267  3255              DCA PTRY                        /PARITY ERROR RETRY COUNTER
0270  1377   SETUP,     TAD (BUFFER-1)                  /PROGRAMS DATA BUFFER
0271  3010              DCA A10                         /AUTO INDEX REGISTER 10
0272  1260              TAD MODE                        / 0 IF 12-BIT, 100 IF 8 BIT
0273  6701              LCD                             /IOT TO ISSUE THE COMMAND
                        /
                        /WAIT FOR A TRANSFER REQUEST FLAG BEFORE TRANSFERRING DATA FROM THE PROGRAMS
                        /
                        /DATA BUFFER TO THE RX01 SECTOR BUFFER
                        /
                        /WAIT FOR A DONE FLAG TO INDICATE THE COMPLETION OF THE FILL BUFFER COMMAND PRIOR TO
                        /
                        /TESTING THE ERROR FLAG
                        /
0274  6703   LOOP,      STR                             /TEST FOR TR FLAG
0275  7410              SKP                             /TR NOT SET, TEST FOR DONE FLAG
0276  5306              JMP FILL                        /TR FLAG SET
0277  6705              SDN                             /TEST FOR DONE FLAG
0300  5274              JMP LOOP                        /NOT TR, OR DONE YET
                        /
                        /THE DONE FLAG IS SET
                        /
                        /TEST FOR ANY ERRORS (ONLY ERROR POSSIBLE IS A PARITY ERROR)
                        /
0301  6704               SER                            /TEST FOR THE ERROR FLAG
0302  7402               HLT                            /NO ERRORS - OK
                        /
                        /INCREMENT AND TEST THE PARITY ERROR RETRY PROGRAM LOCATION " PTRY "
                        /
                        /AND RETRY THE COMMAND UNTIL THE ERROR RECOVERS
                        /
                        /OR UNTIL THE PTRY COUNTER OVERFLOWS TO 0
                        /
0303  2255               ISZ PTRY
0304  5270               JMP SETUP                       /RETRY TO FILL THE SECTOR BUFFER
0305  7402               HLT                             /HARD PARITY ERROR
                        /
                        /THE TRANSFER REQUEST FLAG IS SET
                        /
                        /TRANSFER DATA FROM THE PROGRAMS DATA BUFFER TO THE RX01 SECTOR BUFFER
                        /
0306  1410   FILL,      TAD I A10                       /VIA AUTO INDEX REGISTER 10
0307  6702               XDR                             /TO THE RX01 SECTOR BUFFER
0310  7200               CLA                             /CLA BECAUSE IOT XDR DOESN'T
0311  5274               JMP LOOP                        /LOOP UNTIL THE DONE FLAG SETS
```

Figure 4-13 RX8E Fill Buffer Example

```
                /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    3277            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    WAITSR, 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    6742            XDR                     /LOAD THE SECTOR
0232    7207            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    1360            TAD     TRACK           /0 TO 114 (OCTAL)
0236    6742            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    6742            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 ERROR
0252    2277            ISZ     CTRY            /KEEP COUNT OF RETRIES IF = 12 THEN SKIP
0253    5206            JMP     RETRY           /IF RETRIES <10 THEN DO IT AGAIN
0254    7402            HLT                     /HALT
```

MA-1671

Figure 4-14 RX28 Write/Write Deleted Data/Read Example (Sheet 1 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  7403            BZA                     /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 "FUNCTN" CONTAINS 4 IF WRITE, 14 IF WRITE DELETED DATA, OR
                        /6 IF READ FUNCTION
                        /
0274  0000    FUNCTN, 0
                        /
                        /LOCATION "DRIVEP" HAS BIT ASSIGNMENTS
                        /
                        /       0=      UNIT 0 SINGLE DENSITY
                        /       20=     UNIT 1 SINGLE DENSITY
                        /       40=     UNIT 0 DOUBLE DENSITY
                        /       60=     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
```

MA-1872

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 K10                         / 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    6751            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    6753            STR                             /IOT TO SKIP ON TRANSFER READY
        0324    5323            JMP     .-1                     /IF TR NOT SET LOOP UNTIL IT DOES
        0325    6752            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    6753    LOOP,   STR                             /TEST FOR TR FLAG
        0330    7410            SKP                             /TR NOT SET, TEST FOR DONE FLAG
        0331    5341            JMP FILL                        /TR FLAG SET
        0332    6755            SDN                             /TEST FOR DONE FLAG
        0333    5327            JMP LOOP                        /NOT TR, OR DONE YET
                        /
                        /THE DONE FLAG IS SET
                        /
                        /TEST FOR ANY ERRORS
                        /
        0334    6754            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    6752            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 K10                 / 8 TRYS TO EMPTY THE SECTOR BUFFER
0346  3272            DCA ETRY                /ERROR RETRY COUNTER
0347  1377    ESETJP, TAD (BUFFER-1)          /PROGRAMS DATA BUFFER
0350  3210            DCA A10                 /AUTO INDEX REGISTER 10
0351  1273            TAD MODE                / 0 IF 12-BIT, 100 IF 8 BIT
0352  1274            TAD FUNCJN              / 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  6771            LCD                     /ISSUE COMMAND TO RX
0357  1276            TAD     COMMAND         /GET SAVED COMMAND
0360  7267            AND     4100            /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  6773            STR                     /IOT TO SKIP ON TRANSFER READY
0370  5367            JMP     ..-1            /IF TR NOT SET LOOP UNTIL IT DOES
0371  6772            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  0000
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  6773    ELOOP,  STR                     /TEST FOR TR FLAG
0401  7410            SKP                     /TR NOT SET, TEST FOR DONE FLAG
0402  5212            JMP EMPTY               /TR FLAG SET
0403  6775            SDN                     /TEST FOR DONE FLAG
0404  5200            JMP ELOOP               /NOT TR, OR DONE YET
                        /
                        /THE DONE FLAG IS SET
                        /
                        /TEST FOR ANY ERRORS
                        /
0405  6774            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 ESETJP              /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  6772    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
                      $
```

MA-1874

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.

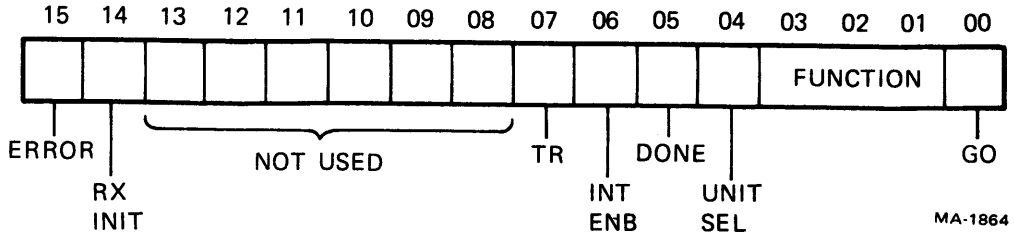


Figure 4-17 RXCS Format (RX11, RXV11)

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.

Bit No.	Description
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.

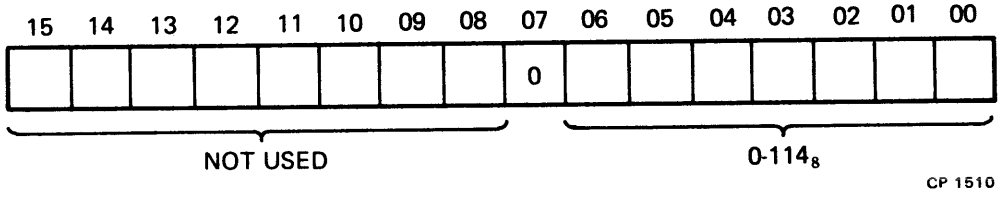


Figure 4-18 RXTA Format (RX11/RXV11)

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.

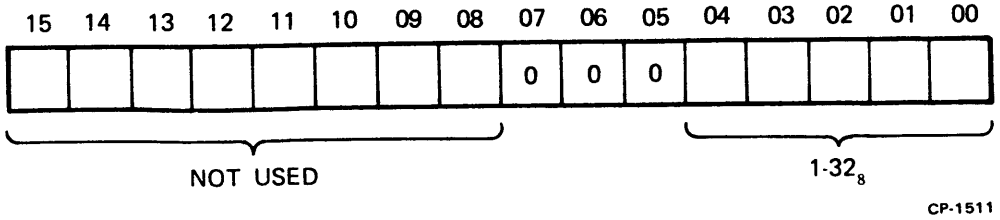


Figure 4-19 RXSA Format (RX11/RXV11)

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).

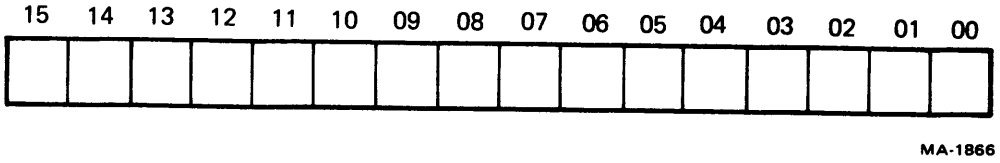


Figure 4-20 RXDB Format (RX11/RXV11)

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.

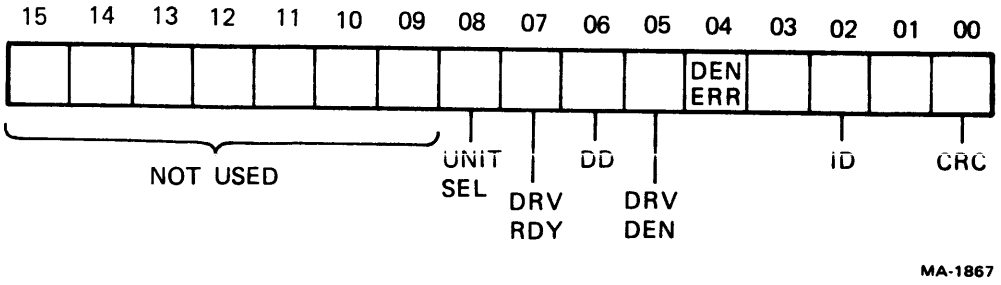


Figure 4-21 RXES Format (RX11, RXV11)

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	

Bit No.	Description
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.

```
                                .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
                               ;
                               ;WAIT FOR THE DONE FLAG AND CHECK FOR ANY ERRORS
                               ;
                               ;IF THE FUNCTION HAS COMPLETED SUCCESSFULLY (NO ERROR FLAG) THEN HALT
                               ;
000060  032767  000040  177102 3$:      BIT #DONEBIT, RXCS       ; TEST FOR THE DONE FLAG
000066  001774                        BEQ 3$                    ; BEQ UNTIL THE DONE FLAG SETS
000070  005767  177074                TST RXCS                  ; TEST FOR THE ERROR FLAG
000074  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
000110  032767  000002  177054        BIT #2, RXES              ; TEST FOR PARITY ERROR
000116  001404                        BEQ CRC                   ; NOT A PARITY ERROR (MUST) BE A CRC
                               ;
                               ;A PARITY ERROR HAS OCCURED
                               ;
                               ;INCREMENT AND TEST THE PARITY ERROR RETRY COUNTER PROGRAM LOCATION " PTRY "
                               ;
                               ;AND RETRY THE " COMMAND " UNTIL THE PARITY ERROR RECOVERS
                               ;
                               ;OR UNTIL THE PTRY COUNTER OVERFLOWS TO 0
                               ;
000120  005267  000202                INC PTRY
000124  001336                        BNE RETRY                 ; RETRY THE COMMAND
000126  000000                        HALT                      ; HARD PARITY ERROR
                               ;
                               ;A CRC ERROR HAS OCCURED
                               ;
                               ;INCREMENT AND TEST THE CRC ERROR RETRY COUNTER PROGRAM LOCATION " CTRY "
                               ;
                               ;AND RETRY THE COMMAND UNTIL THE CRC ERROR RECOVERS
                               ;
                               ;OR UNTIL THE CTRY COUNTER OVERFLOWS TO 0
                               ;
000130  005267  000174        CRC:     INC CTRY
000134  001332                        BNE RETRY                 ; RETRY THE COMMAND
000136  000000                        HALT                      ; HARD CRC ERROR
                               ;THE ERROR FLAG IS SET
                               ;
                               ;THE ERROR IS (NOT) A PARITY ERROR AND IS (NOT) A CRC ERROR
                               ;
                               ;THEREFORE IT MUST BE A SEEK ERROR
                               ;
                               ; (STATE OF RXCS BITS 0 AND 1 ARE 0)
                               ;
000140  012767  040000  177022 SEEK:   MOV #INIT, RXCS          ; INITIALIZE
                               ;
                               ;INCREMENT AND TEST THE SEEK ERROR RETRY COUNTER PROGRAM LOCATION " STRY "
                               ;
                               ;AND RETRY THE COMMAND UNTIL THE SEEK ERROR RECOVERS
                               ;
                               ;OR UNTIL THE CTRY COUNTER OVERFLOWS TO 0
                               ;
000146  005267  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

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.

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.

```
160                                     ;THE FOLLOWING IS A PROGRAMMING EXAMPLE OF PROTOCOL REQUIRED TO
161                                     ;
162                                     ;EMPTY THE  SECTOR BUFFER OF 128 8-BIT BYTES
163                                     ;
164  000242  012767  177770  000056     EENTRY: MOV #-10, PTRY          ; 8 TRYS TO EMPTY THE SECTOR BUFFER
165  000250  012700  000342             ESETUP: MOV #BUFFER, R0         ; PROHGRAMS DATA BUFFER
166  000254  016767  000054  176706             MOV COMMAND, RXCS       ; ISSUE THE COMMAND
167                                     ;
168                                     ;WAIT FOR A TRANSFER REQUEST FLAG BEFORE TRANSFERRING DATA TO THE PROGRAMS
169                                     ;
170                                     ;DATA BUFFER FROM THE RX01 SECTOR BUFFER
171                                     ;
172                                     ;WAIT FOR A DONE FLAG TO INDICATE THE COMPLETION OF THE EMPTY BUFFER COMMAND
173                                     ;
174                                     ;PRIOR TO TESTIN G THE ERROR FLAG
175                                     ;
176  000262  105767  176702             ELOOP:  TSTB RXCS               ; TEST FOR TRANSFER REQUEST FLAG
177  000266  001014                             BMI EMPTY               ; BNE IF TRANSFER REQUEST FLAG IS SET
178  000270  032767  000040  176672             BIT #DONEBIT, RXCS      ; TEST FOR DONE FLAG
179  000276  001771                             BEQ ELOOP               ; BEQ UNTIL THE DONE FLAG SETS
180                                     ;
181                                     ;THE DONE FLAG IS SET
182                                     ;
183                                     ;TEST FOR ANY ERRORS (ONLY ERROR POSSIBLE IS A PARITY ERROR)
184                                     ;
185  000300  005767  176664                     TST RXCS
186  000304  001001                             BNE 1$
187  000306  000000                             HALT                    ; NO ERRORS = OK = COMPLETE
188                                     ;
189                                     ;INCFEMENT AND TEST THE PARITY ERROR RETRY PROGRAM LOCATION " PTRY "
190                                     ;
191                                     ;AND RETRY THE COMMAND UNTIL THE ERROR RECOVERS
192                                     ;
193                                     ;OR UNTIL THE PTRY CUNTER OVERFLOWS TO 0
194                                     ;
195  000310  005267  000012             1$:     INC PTRY
196  000314  001355                             BNE ESETUP              ; RETRY TO EMPTY THE SECTOR BUFFER
197  000316  000000                             HALT                    ; HARD PARITY ERROR
198                                     ;
199                                     ;THE TRANSFER REQUEST FLAG IS SET
200                                     ;
201                                     ;TRANSFER DATA TO THE PHOGRAM  DATA BUFFER FROM THE RX01 SECTOR BUFFER
202                                     ;
203  000320  116730  176646             EMPTY:  MOVB RXDB, @(R0)+
204  000324  000756                             BR ELOOP
205                                     ;THE FOLLOWING 3 PROGRAM LOCATIONS ARE THE ERROR RETRY COUNTERS
206                                     ;
207  000326  000000                     PTRY:   0                       ; PARITY ERROR RETRY COUNTER
208  000330  000000                     CTRY:   0                       ; CRC ERROR RETRY COUNTER
209  000332  000000                     STRY:   0                       ; SEEK ERROR RETRY COUNTER
210                                     ;
211                                     ;PROGRAM LOCATION " COMMAND " CONTAINS THE COMMAND TO BE ISSUED VIA THE LCD IOT
212                                     ;
213                                     ;WRITE (4), WRITE DELETED DATA (14), OR READ (6), OR EMPTY BUFFER (2)
214                                     ;
215  000334  000000                     COMMAND:        0               ; 4, 14, 6, OR 2 + (GO BIT 1 = 1)
216                                     ;
217                                     ;PROGRAM LOCATION " SECTOR " CONTAINS THE SECTOR ADDRESS (1 TO 32 OCTAL)
218                                     ;
219  000336  000000                     SECTOR: 0                       ; 1 TO 32 OCTAL
220                                     ;
221                                     ;PROGRAM LOCATION " TRACK " CONTAINS THE TRACK ADDRESS (0 TO 114 OCTAL)
222                                     ;
223  000340  000000                     TRACK:  0                       ; 0 TO 114 OCTAL
224                                     ;
225                                     ;PROGRAM EQUIVALENTS
226                                     ;
227           000040                    DONEBIT=40
228           040000                    INIT=40000
229           000342                    BUFFER=.
230           000542                            .=BUFFER+200
231           000001                            .END
232
```

Figure 4-23 RX11/RXV11 Empty Buffer Example

```
111                                     ;THE FOLLOWING IS A PROGRAMMING EXAMPLE OF THE PROTOCOL REQUIRED TO
112                                     ;
113                                     ;FILL THE SECTOR BUFFER WITH 128 8-BIT BYTES
114                                     ;
115                                     ; NOTE: THE DATA TO FILL THE SECTOR BUFFER CAN BE ASSEMBLED IN CORE IN THE
116                                     ;       EVEN ADDRESSES BYTES OF 128 WORDS OR IN BOTH BYTES OF 64 WORDS
117                                     ;
118  000156  012767  177770  000142   FENTRY: MOV #-10, PTRY          ; 8 TRYS TO FILL THE SECTOR BUFFER
119  000164  012700  000342           SETUP:  MOV #BUFFER, R0         ; PROGRAMS DATA BUFFER
120  000170  016767  000140  176772           MOV COMMAND, RXCS       ; ISSUE THE COMMAND
121                                     ;
122                                     ;WAIT FOR A TRANSFER REQUEST FLAG BEFORE TRANSFERRING DATA FROM THE PROGRAMS
123                                     ;
124                                     ;DATA BUFFER TO THE RX01 SECTOR BUFFER
125                                     ;
126                                     ;WAIT FOR A DONE FLAT TO INDICATE THE COMPLETION OF THE FILL BUFFER COMMAND
127                                     ;
128                                     ;PRIOR TO TESTING THE ERROR FLAG
129                                     ;
130  000176  105767  176766           LOOP:   TSTB RXCS               ; TEST FOR TRANSFER REQUEST FLAG
131  000202  001414                           BMI FILL                ; BEQ IF TRANSFER REQUEST FLAG SET
132  000204  032767  000040  176756           BIT #DONEBIT, RXCS      ; TEST FOR THE DONE FLAG
133  000212  001771                           BEQ LOOP                ; BEQ UNTIL THE DONE FLAG SETS
134                                     ;
135                                     ;THE DONE FLAG IS SET
136                                     ;
137                                     ;TEST FOR ANY ERRORS (ONLY ERROR POSSIBLE IS A PARITY ERROR)
138                                     ;
139  000214  005767  176750                   TST RXCS
140  000220  001001                           BNE 1$
141  000222  000000                           HALT                    ; NO ERRORS = OK = COMPLETE
142                                     ;
143                                     ;INCREMENT AND TEST THE PARITY ERROR RETRY PROGRAM LOCATION " PTRY "
144                                     ;
145                                     ;AND RETRY THE COMMAND UNTIL THE ERROR RECOVERS
146                                     ;
147                                     ;OR UNTIL THE PTRY COUNTER OVERFLOWS TO 0
148                                     ;
149  000224  005267  000076           1$:     INC PTRY
150  000230  001355                           BNE SETUP               ; RETRY TO FILL THE SECTOR BUFFER
151  000232  000000                           HALT                    ; HARD PARITY ERROR
152                                     ;
153                                     ;THE TRANSFER REQUEST FLAG IS SET
154                                     ;
155                                     ;TRANSFER DATA FROM THE PROGRAMS DATA BUFFER TO THE RX01 SECTOR BUFFER
156                                     ;
157  000234  113067  176732           FILL:   MOVB @(R0)+, RXDB       ; PROGRAMS DATA BUFFER IS 64 WORDS IN LENGTH
158  000240  000756                           BR LOOP
```

Figure 4-24 RX11/RXV11 Fill Buffer Example

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.

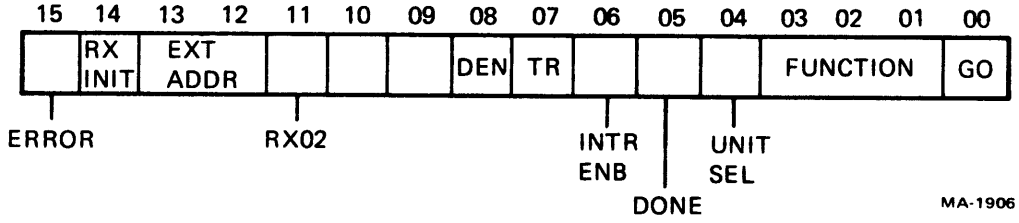


Figure 4-25 RX2CS Format (RX211/RXV21)

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.

Bit No.	Description
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 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.

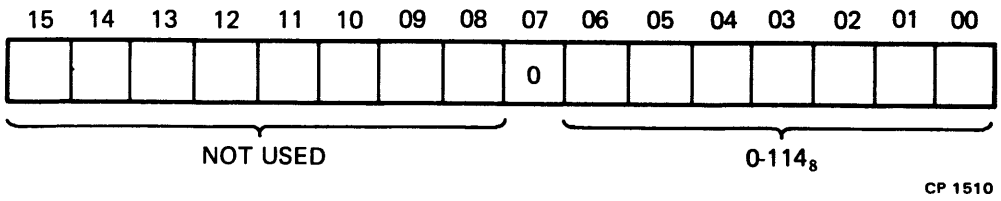


Figure 4-26 RX2TA Format (RX211/RXV21)

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

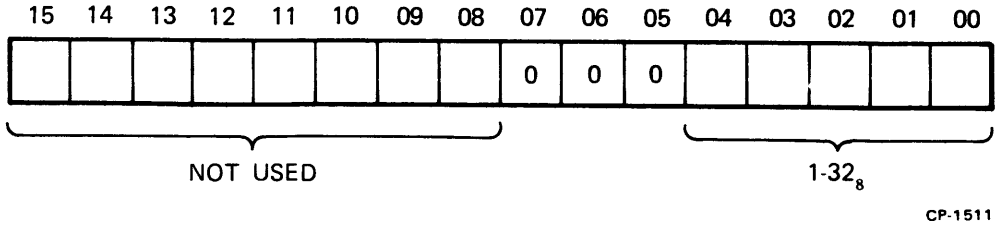


Figure 4-27 RX2SA Format (RX211/RXV21)

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.

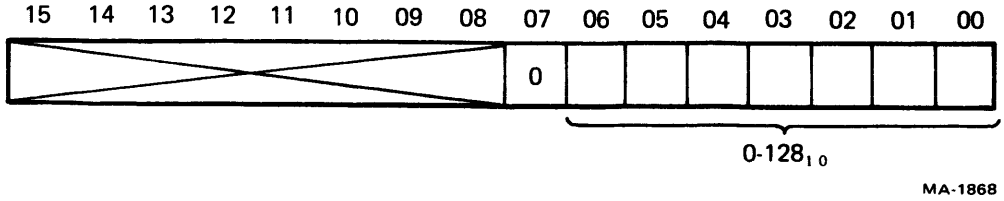


Figure 4-28 RX2WC Format (RX211/RXV21)

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).

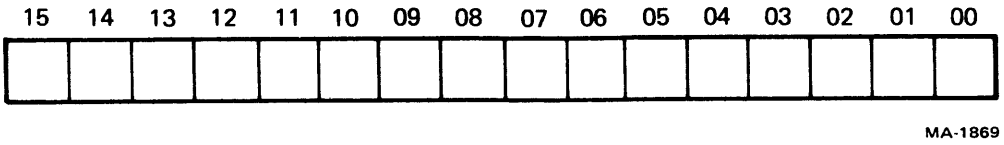


Figure 4-29 RX2BA and RX2DB Format (RX211/RXV21)

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.

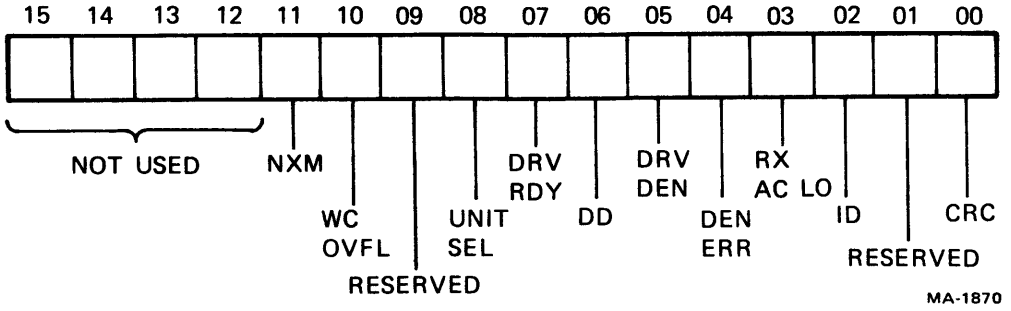


Figure 4-30 RX2ES Format (RX211/RXV21)

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.

Bit No.	Description
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 theword 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>	Unnit 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	†

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.

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.

---

* 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 reacheed on a seek error.

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.

4.3.4 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 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. 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/RXV21 Programming 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  001070          OUTPUT: TST     FIN             ;IF FINI FLAG
001172  001041                          BNE     ENDOUT          ;EQUALS ZERO THEN
001174  004767  000100                  JSR     PC,OUBUF2       ;FILL RX02 BUFFER
001200  000240                          NOP
001202  016767  001032  001026          MOV     UTI,CMD         ;SELECT DRIVE
001210  052767  000105  001020          BIS     #105,CMD        ;SET TO WRITE SECTOR + INT ENABLE
001216  056767  001052  001012          BIS     DENSITY,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     DENSITY,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  000738                  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
                                ;*****************************************************************

001400  000240                  INPUT:  NOP
001410  016767  000624  000620          MOV     UIT,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     TP,@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  052767  000504  000524          BIS     DENSITY,CMD     ;SET DENSITY
001512  016777  000520  000610          MOV     CMD,@RXCS       ;ELSE LOAD COMMAND
001520  004767  000422                  JSR     PC,AWTR         ;WAIT FOR "TR"     DO MOD 0,TR
001524  005767  000540                  TST     FIN             ;IF FINI FLAG
001530  001014                          BNE     ENDIN2          ;EQUALS ZERO
001532  016777  000530  000572          MOV     #DCNT,@RXDB     ;THEN LOAD WORD COUNT
001540  004767  000462                  JSR     PC,AWTR         ;WAIT FOR "TR"     DO MOD 0,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 0,WTR - AWAIT TRANSFER READY SUBROUTINE
                                ;*****************************************************************

002146  005267  000060          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-1882
```

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

CHAPTER 5
THEORY OF OPERATION

This chapter describes the functional operation of the hardware and μCPU software of the RX02 Floppy Disk System. This information combined with the programming information and descriptions contained in the other chapters should give the reader an understanding of the RX02 Floppy Disk System theory of operation.

The first section of this chapter describes the overall system block diagram and the interconnection between system assemblies; the second section describes the interface modules; and the third section describes the assemblies housed within the RX02 cabinet. (Reference should be made to the *PDP-11 Peripherals Handbook*, the *Microcomputer Handbook* and the *PDP-8 Small Computer Handbook* for input/output transfer information for each computer.)

5.1 OVERALL SYSTEM BLOCK DIAGRAM

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

1. Drive mechanics, which includes actuators and transducers (up to two per controller) to read/write data.

2. Read/write electronics, which transfers control and data signals between drive mechanics and control logic.

3. Microprogrammed controller, which includes all control logic.

4. Bus interface module, which transfers control and data signals between the host processor bus and the RX02 microprogrammed controller.

Operation of the RX02 Floppy Disk System is governed by the host processor. General functions (such as write data, read data) to be performed are initiated by the processor and coupled through the interface module to the μCPU controller. The μCPU decodes the general function, and using its own microprogram, initiates specific functions to accomplish the general function. The μCPU output governs the read/write electronics which in turn develop the necessary signals to position the read/write heads in the desired location in order to read or write data. The data paths between each of the elements is bidirectional so that for reading, the data flow is from the disk to the read/write electronics, to the μCPU, to the interface, and then to the host processor; for writing, the data flow is reversed.

5.1.1 Omnibus to RX8E/RX28 Interface Signals

The RX8E/RX28 interface communicates with the PDP-8 Omnibus via the signals shown in Figure 5-2 and described below.

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

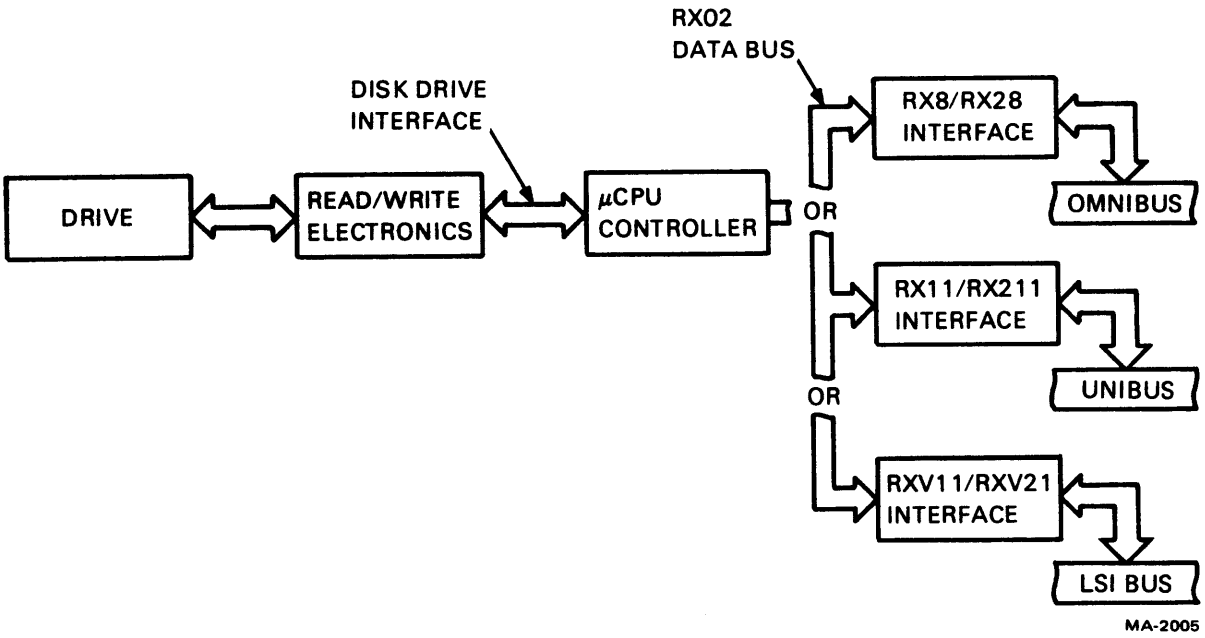


Figure 5-1 RX02 System Block Diagram

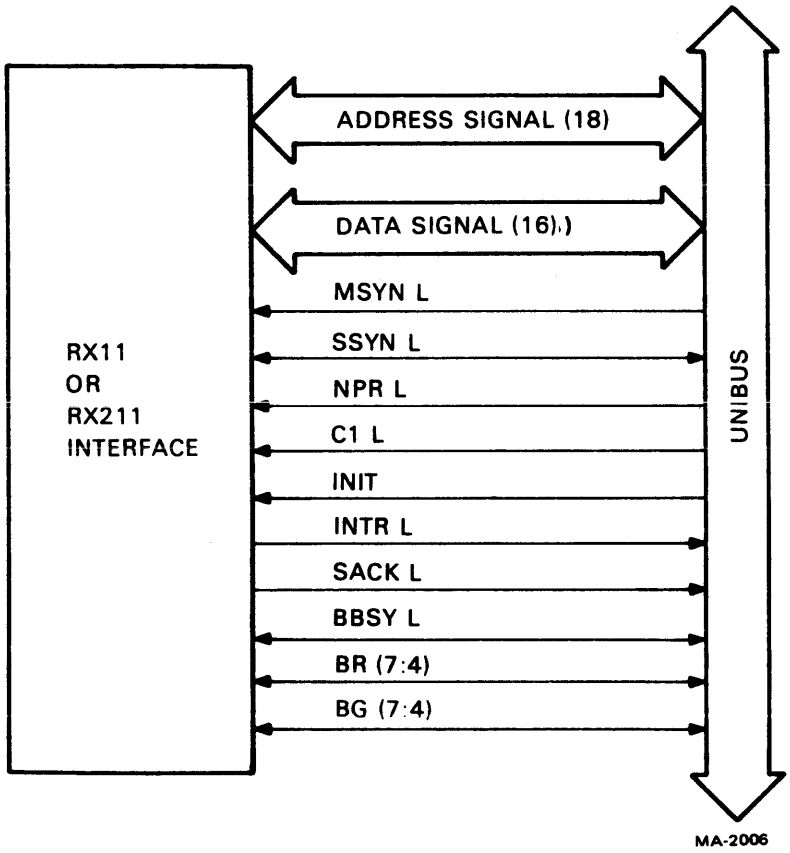


Figure 5-2 Omnibus to RX8E/RX28 Interface Signals

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

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

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

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

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

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

INIT H – INIT H is a signal used to clear all flags in the interface and initialize the RX02.

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

5.1.2 Unibus to RX11/RX211 Interface Signals

The RX11/RX211 interface communicates with the PDP-11 Unibus via the signals shown in Figure 5-3 and described below.

ADDRESS (A Lines) – The 18 address lines are used by the CPU to select the device register addresses of the RX11/RX211 which are 177170 (RXCS) and 177172 (RXDB).

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

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

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

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

NOTE
The RX11 is not an NPR device. The RX211 is an NPR device.

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

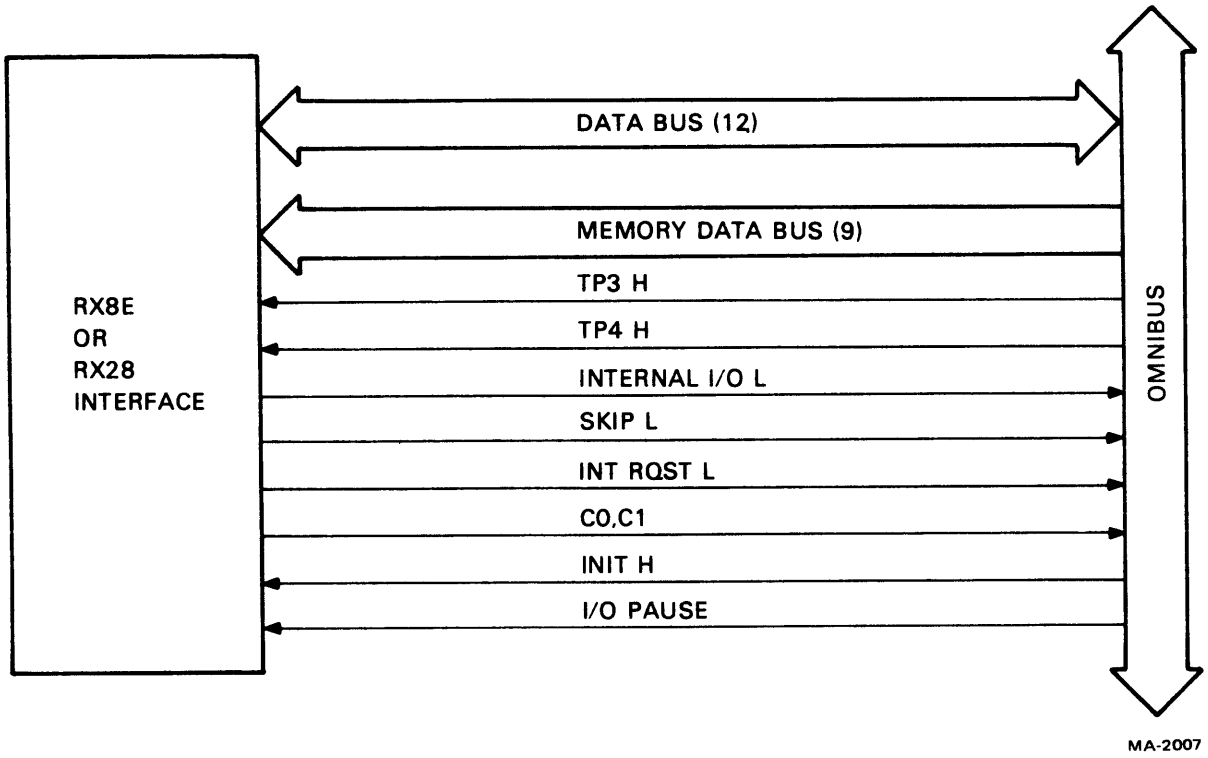


Figure 5-3 Unibus to RX11/RX211 Interface Signals

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

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

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

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

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

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

5.1.3 LSI-11 Bus to RXV11/RXV21 Interface Signals

The RXV11 or RXV21 interface module communicates with the LSI-11 via signals shown in Figure 5-4 and described below.

BDAL0:15L (Data/Address) – The 16 data/address lines are used by the CPU to communicate data and address information in and out of the interface module.

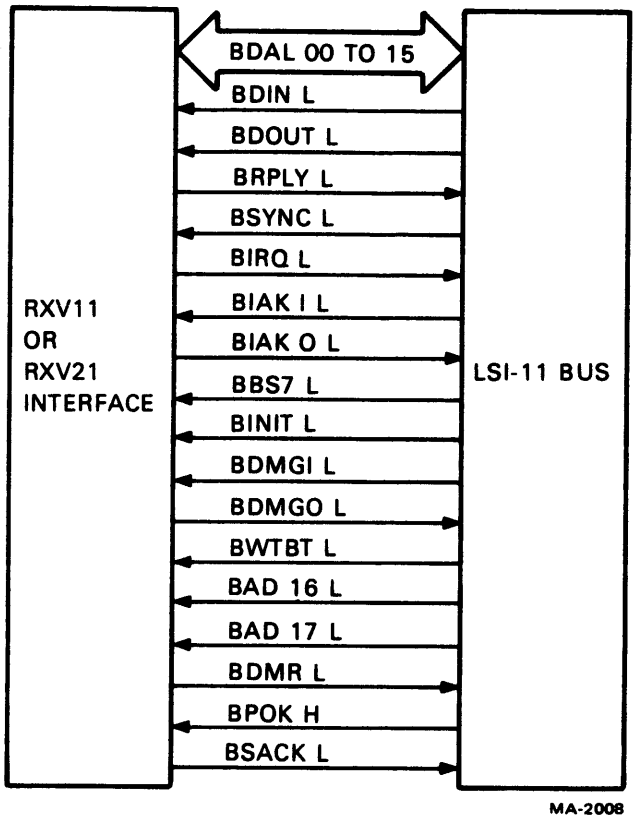


Figure 5-4 LSI-11 Bus to RXV11/RXV21 Interface Signals

BDIN L – This signal is used with BSYNCL to indicate that the CPU is ready to accept data. When used without BYSNCL, it is used to indicate that the CPU has an interrupt in progress.

BDOUT L – This signal is used by the CPU to indicate that valid data is available on BDAL 0–15.

BRPLY L – This signal is asserted by the interface module to indicate to the CPU that it has either accepted data from the bus or placed data on the bus.

BSYNC L – This signal is the master synchronization control signal asserted by the master device when it has control of the bus for transmitting an address.

BIRQ L – The interface asserts this signal to indicate to the CPU that it has data to be processed.

BIAK IL, BIAK OL – BIAK IL, interrupt acknowledge in, is asserted by the CPU in response to a BIRQL. If the interface is not asserting BIRQL, it will pass the signal via BIAK OL to the next device.

BBS7 L – The master device asserts this signal when an address in the 28K–38K range is placed on the bus.

BINIT L – This signal is asserted by the CPU to initialize or clear the interface module as well as all devices connected to the I/O bus.

BDMGI L, BDMGO L – BDMGI L, DMA grant-input, is asserted by the CPU and routed to the first device on the bus. If the device is not requesting the bus, the signal is routed to the next device via BDMGO L.

BWTBT L – This signal, write/byte, is asserted during BSYNC L to indicate that an output sequence is to follow; it is also asserted during BDOUT L for byte addressing.

BAD16 L, BAD17 L – Extended address bits.

BDMRL – This signal, direct memory access request, is asserted by the interface to request bus mastership.

BPOK H – This signal indicates the processor power supply operation is normal.

BSACK L – This signal is asserted by the interface to indicate it is bus master.

5.1.4 Interface Module to μCPU Controller Signals

The μCPU controller and interface modules communicate via the signals shown in Figure 5-5 and described below.

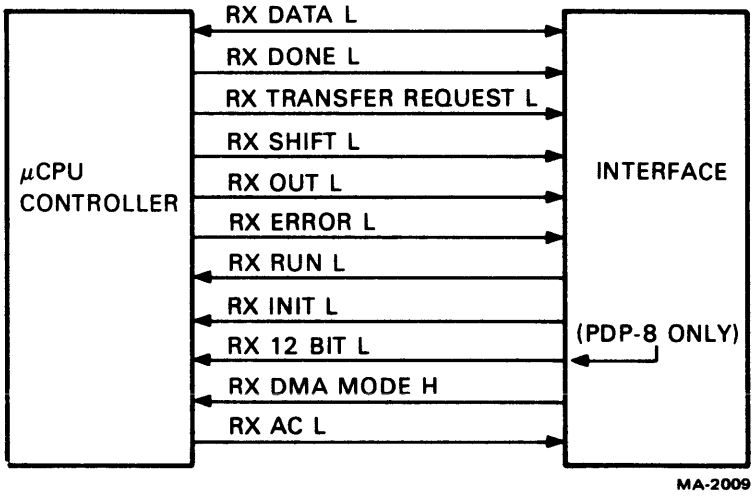


Figure 5-5 Interface to μCPU Controller Signals

RX INIT L – The RX02 will negate DONE L and move the head position mechanism of drive 1 (if it exists) to track 0. The RX02 will also read sector 1 of track 1 of drive 0 and then assert DONE L without error upon successful completion of the function.

RX DONE L – This signal, asserted low, indicates that there is no RX02 function in progress. Initiating any function will cause DONE L to be negated for the duration of that function. Attempting to initiate any function other than Initialize while DONE L is negated is illegal and may result in an error.

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

RX OUT L – This signal indicates the direction in which the RX02 is prepared to transfer data. When OUT L is asserted, the direction of transfer is from controller to interface. When OUT L is negated, the direction is from interface to controller. OUT L is never asserted while DONE L is true, and OUT L is negated by Initialize.

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

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

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

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

1. Interface to Controller Transfer – OUT is negated and the transfer of either commands or data words begins with the assertion of RUN L. Following the assertion of RUN L, DONE L or TR L will be negated and a number of SHIFT L pulses will occur. The number depends on the length of the data element to be passed.

   The first bit of data (or command) must be stable when RUN L is asserted. The SHIFT L pulse width is 200 ns nominal. SHIFT L pulses will not occur more often than every 400 ns. Subsequent bits of data may be brought up on the trailing edge of SHIFT L. DONE L or TR L will be reasserted following the last SHIFT L pulse.

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

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

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

NOTE

Signal 12 Bit L is only asserted by a PDP-8 interface for 12-bit words. It is never asserted by a PDP-11 or LSI-11 interface.

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

RX DMA MODE H – This line is asserted to indicate direct memory access mode of operation.

RX AC L – This line is asserted when the RX02 has lost power.

5.1.5 μCPU Controller to Read/Write Electronics Signals

The μCPU controller and read/write electronics communicate via the signals shown in Figure 5-6 and described below.

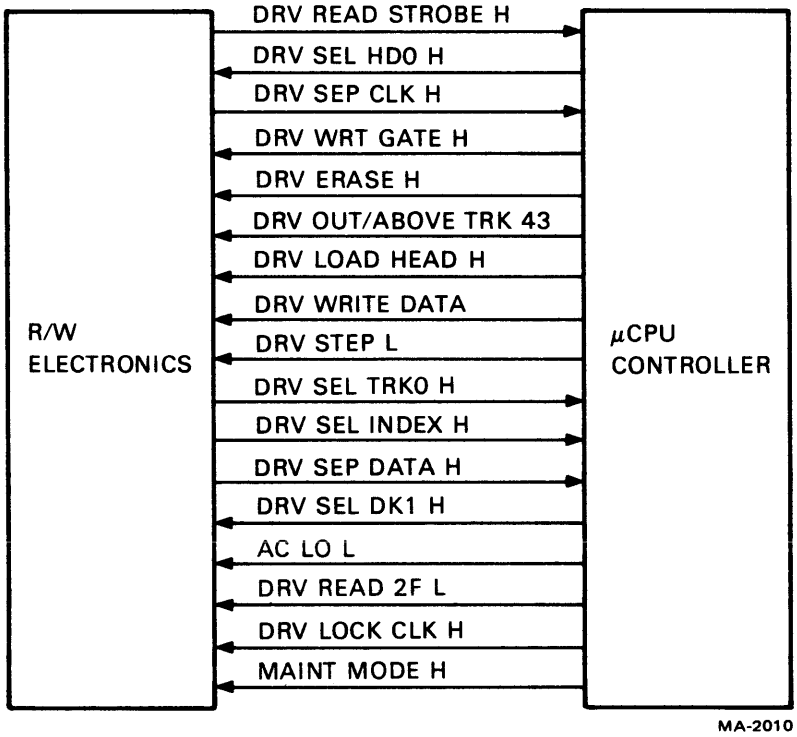


Figure 5-6 μCPU Controller to Read/Write Electronics Signals

DRV OUT/ABOVE TRK 43– When writing data, this signal is asserted by the controller to select the lower of two write current levels when operating on a track above 43. As the head moves closer to the center of the disk, the bit density increases as linear velocity decreases, necessitating a reduction in write current. When reading data, this signal determines the direction in which the head will move in response to a STEP L signal; when OUT/ABOVE TRK 43 is asserted, the heads will travel toward the outer edge of the disk.

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

DRV SEP DATA H – This signal conveys the complete data stream recovered from the diskette. It is asserted with READ STROBE H and remains valid for 4 μs for single density data and 2 μs for double density data.

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

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

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

DRV LOAD HEAD H – This signal is asserted by the controller to hold the media against the selected head.

DRV STEP L – This signal is asserted twice by the controller to change head position by one track in a direction determined by signal OUT H. There are two step pulses per track for each track to be stepped (6 ms step time per track).

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

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

AC LO L – This signal is asserted by an Initialize signal from the controller to the drives, during and after a loss of power (an initialize is provided over this line).

DRV READ STROBE H – This signal is asserted by the selected drive to indicate that SEP DATA H and SEP CLK H are valid and may be stored. This signal is asserted for approximately 1 μs every 4 μs for single density and every 2 μs for double density.

DRV SEL HD O H – This line (for future use) selects one of two possible heads on a disk drive.

DRV SEP CLK H – This line indicates that a separated clock has been detected in a data stream. SEP CLOCK H is asserted with READ STROBE H and remains valid for 4 μs for single density and for 2 μss for double density.

DRV READ 2 F L – This signal controls the density at which the phase lock loop (PLL) will acquire and/or maintain phase lock.

DRV LOCK CLK H – This signal controls the phase lock loop (PLL) and preamble detection so that the PLL will acquire phase lock during the six bytes of zeros preceding the ID, Data, or deleted data address marks. The PLL does not function when LOCK CLK H is negated.

DRV MAINT MODE H – When asserted, the R/W electronics are exercised without storing or retrieving data from the media. This mode tests the digitized portion of R/W electronics including preamble recognition circuitry, the PLL, and the data separator.

5.1.6 Read/Write Electronics to Drive Signals

The read/write electronics and drive(s) communicate through five sets of signals per drive as shown in Figure 5-7 and described below. The plug designations for the cabling are also shown in Figure 5-7.

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

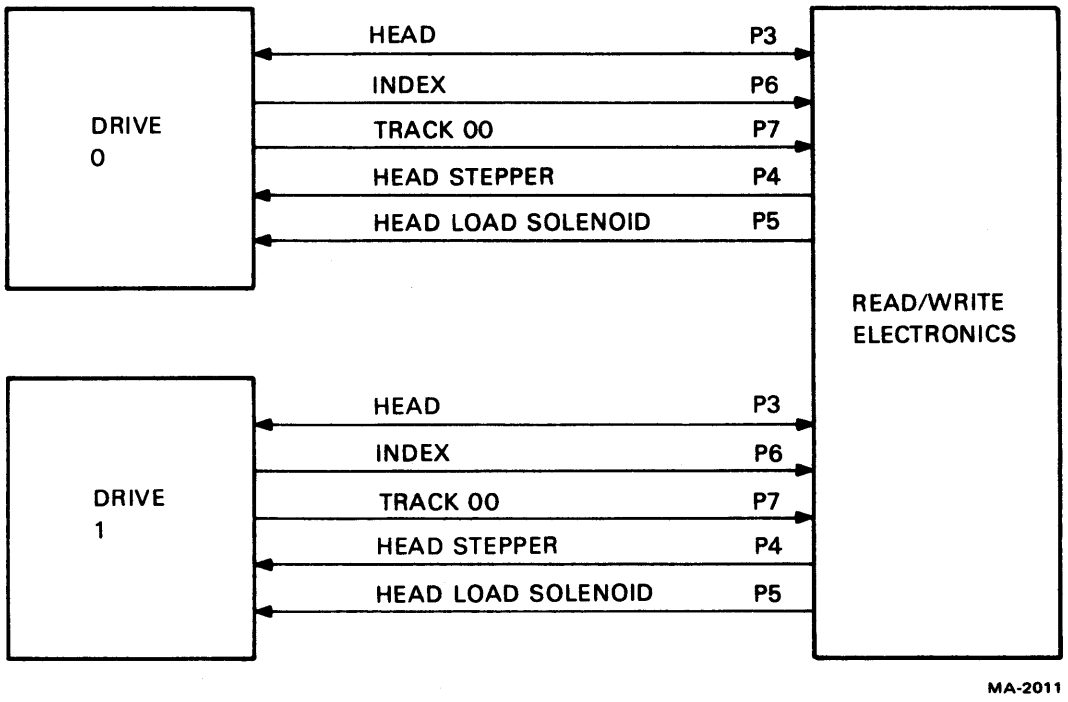


Figure 5-7 Read/Write Electronics to Drive Signals

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

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

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

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

5.2 INTERFACE MODULES BLOCK DIAGRAM DESCRIPTION

The following paragraphs contain functional block diagram descriptions of each of the interface modules used with the RX02.

5.2.1 RX8E/RX28 Interface (M8357) Block Diagram Description

Figure 5-8 presents a block diagram of the RX8E/RX28 interface. The E references on the diagram are IC chip designations on the RX8/RX28 print set, which is a separate document.

5.2.1.1 Device Select and IOT Decoder – The device select and IOT decoder logic decodes instructions from the memory data bus and generates signals to the interrupt control and skip logic, the C line control logic, and the sequence and function control logic.

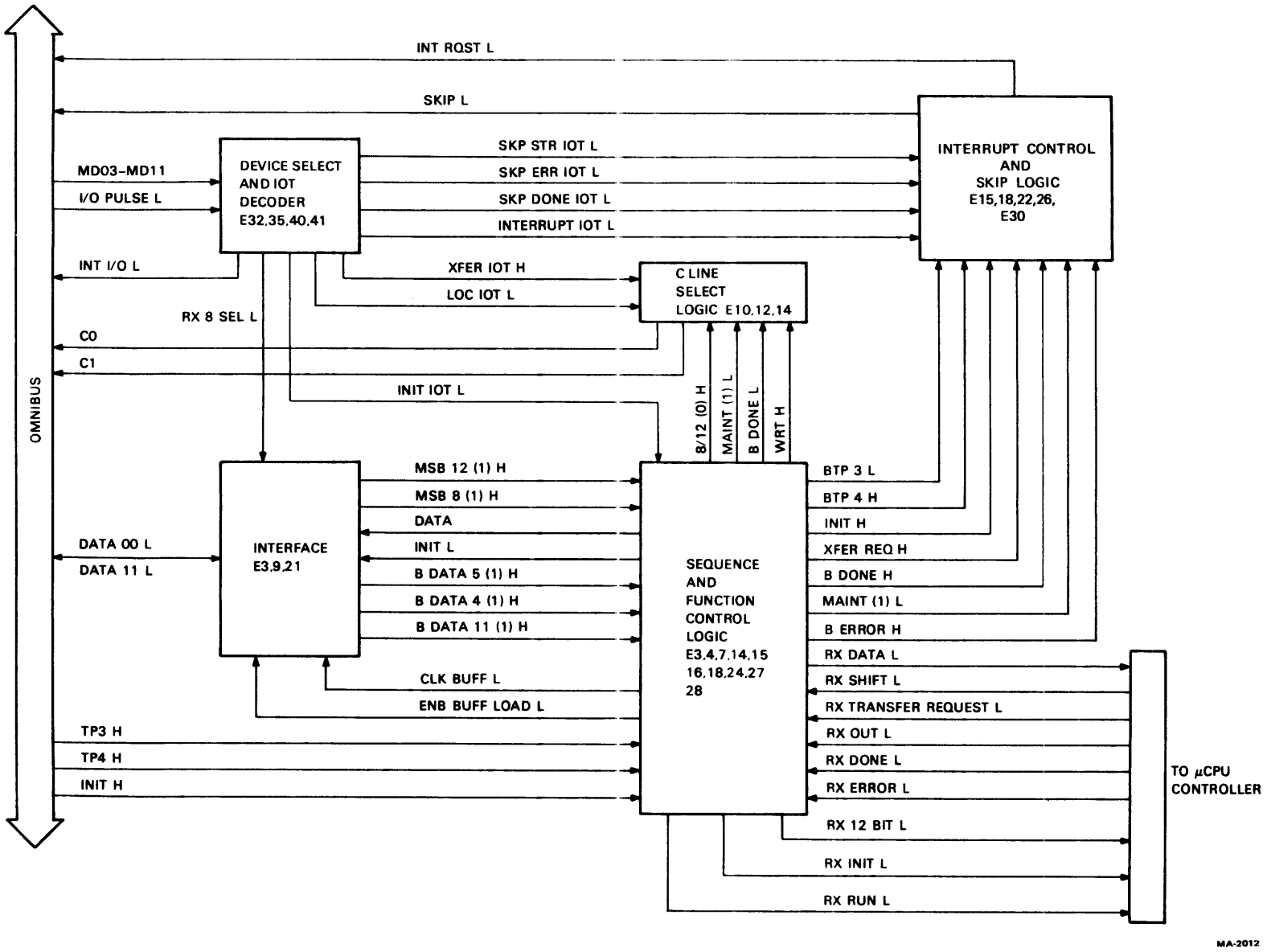


Figure 5-8 RX8E/RX28 Interface Block Diagram

Device selection codes are determined by the switch configuration with relation to the state of MD6, MD7, and MD8. When the correct code for the RX8E/RX28 is input to the device select logic on MD03 L–MD08 L and I/O PAUSE L is asserted, MD09 L–MD11 L are decoded by the decoder E41 and signal INT I/O is asserted on the Omnibus. I/O PAUSE L is present anytime an IOT instruction is being executed by the program. INT I/O L prevents the processor from executing other I/O transfers (IOTs) while this instruction is being executed.

Decoder E41 is a BCD to decimal decoder. All 0s applied to inputs D0–D3 (D3 is MSB) will cause pin 1, which is unused, to be asserted low. An input of 001 will cause signal LCD IOT L to be asserted. An input of 010 (decimal 2) will cause XFER IOT L to be asserted. Therefore, for each function code input on MD09 L–MD11 L, only one of the decoder output lines will be asserted. The function codes are further explained in Paragraph 4.1.2.

5.2.1.2 Interrupt Control and Skip Logic – The interrupt control logic asserts the BUS INT RQST L signal on the Omnibus. Bit 11 of the data bus must be set and an INTERRUPT IOT L must be decoded by the interface to set the interrupt enable flip-flop. The combination of the interrupt enable and buffered done flip-flops will assert BUS INT RQST L. Setting the buffered done flip-flop indicates that no RX02 function is currently in progress.

The skip logic implements the three IOT commands Skip on Transfer Request Flag (STR), Skip on Error Flag (SER), and Skip on Done Flag (SDN) as described in Paragraph 4.1.2.

NOTE
When using these instructions, the respective flags are cleared after they are tested.

Signal SKIP L will be asserted if any of the above instructions are decoded by the IOT decoder and the respective flag has been asserted by the RX02.

The RX8E/RX28 asserts the flags by causing a positive transition on the clock inputs of flip-flops XFER REQ, ERR, and DONE. The signal MAINT (1) L will directly set the skip flags to allow the skip IOTs to assert the BUS SKIP L signal when decoded by the IOT decoder.

5.2.1.3 C Line Select Logic – The C line select logic (E10, E12, E14) controls the direction of data flow between the processor AC and the data bus and determines whether or not the AC is cleared upon completion of the transfer. CO L will be asserted during a load command (LCD) instruction when signal LCD IOT L is asserted. Assertion of C1 L requires XFER IOT H to be asserted and either MAINT (1) L or B DONE L to be asserted or WRT H to be negated. Data transfers occur under the control of the C bits according to Table 5-1.

5.2.1.4 Interface Register – The interface register (E9, E21, E3) is made up of three 8271 4-bit shift registers. This register temporarily stores data during transfers from the Omnibus to the RX02 μCPU controller or during transfers from the μCPU controller to the Omnibus.

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

During data transfer to the Omnibus (Empty Buffer), serial data from the μCPU controller is shifted into the buffer. ENB BUFF LOAD L must be negated while CLK BUFF L supplies the clock pulses.

Table 5-1 C Line Transfer Control Signals

Type of Transfer	C0	C1	Action Required by RX8E/RX28 Interface	Action by Processor
Output AC to data bus; AC unchanged.	H	H	Load data bus into buffer.	Transfers AC to data bus. AC remains unchanged.
Output AC to data bus; AC cleared	L	H	Load data bus into buffer. Ground C0.	Transfers AC to data bus and clears AC.
Input AC-ORed with peripheral data	H	L	Gate peripheral data to data bus. Ground C1.	AC-ORed with peripheral data.
Jam input data bus to AC.	L	L	Gate peripheral data to data bus. Ground C0 and C1.	Transfers data bus to AC register.

The parallel data is enabled from the outputs of the register when MAINT (1) H, RDH, or B DONE H is asserted, and when XFER IOT L is asserted as decoded by the IOT decoder. Only eight bits of data will be output if signal 8/12 (0) H is low; otherwise, 12 bits will be transmitted.

5.2.1.5 Sequence and Function Control Logic - The sequence and function control logic performs six distinct functions:

1. Controls loading and shifting of the interface register to and from the μCPU controller.
2. Senses 8- or 12-bit mode and outputs RX 12 BIT L.
3. Senses maintenance conditions.
4. Generates INIT L signal to the μCPU controller.
5. Generates RUN L signal to the μCPU controller.
6. Generates a parity bit for the serial bit stream to the RX02.

Interface register operation is controlled by signals ENB BUFF LOAD L and CLK BUFF L generated by the sequence and function control logic. To assert ENB BUFF LOAD L, signal RX OUT L cannot be asserted and either RX TRANSFER REQUEST L or RX DONE L must be asserted.

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

1. ENB BUFF LOAD L is asserted
2. Either LCD OR XDR instructions are being performed
3. Signal BUS TP3 H is asserted.

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

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

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

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

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

5.2.2 RX11 Interface (M7846) Block Diagram Description

A block diagram of the RX11 interface is shown in Figure 5-9. The E references are IC chip designations on the RX11 print set, which is a separate document.

5.2.2.1 Address Decoder – The address decoder determines whether this interface is being addressed and whether control information or data is being transferred.

The hardware E20, E24, E27, E30 is a logic network that decodes two discrete addresses assigned to the RX11. Address bits (17:13) must always be asserted to satisfy the decoder. The state of address bits A (12:03) is determined by the placement of jumpers A12–A3 on the board. For each of these bits, one 8242 exclusive-NOR gate is used. Insertion of aa jumper for a particular bit position stores a 0 on one leg of the 8242 so that a 1 appearing on the other leg causes the output to go low. This is a mismatch condition which is met when the associated address bit is a 1. When both legs match (1s or 0s on both), the output is high. The output of these 12 gates are wire-ORed and applied to pin 9 of the E28 gate. Pin 10 of thiss gate receives the NANDed signal of A (17:13) and BUS MSYN. Pin 8, the output of this gate, is signal REG SELECT L and is asserted when the proper Unibus addresses are decoded.

The states of address bits A02 and A01 and REG SELECT L are used to produce signals SELECT 00 H or SELECT 02 H. If BUS A01 L is asserted, SELECT 02 H is asserted. If BUS A01 L is negated, SELECT 00 H is asserted. These signals, in turn, allow access to the RXCS register when SELECT 00 H is asserted, or to the RXDB register when SELECT 02 H is asserted. (Refer to Paragraph 4.2 for register descriptions.)

5.2.2.2 Data Path Selection – Data path selection logic provides the input/output path for data. Signal BUS C1 L from the Unibus controls whether a Data Out or Data In operation is to be executed. Assertion of BUS C1 L indicates Data Out (from Unibus), and negation of BUS C1 L indicates Data In (to Unibus). Signals OUT H and its complement IN H are derived from BUS C1 L and are input to bus transceivers E4 and E7. With BUS C1 L negated and SELECT 02 asserted, all eight bits from the data buffer are enabled through the bus transceivers. With BUS C1 L negated and SELECT 00 H asserted, only lines BUS D04–BUS D07, providing control and status information (RXCS), are enabled. With BUS C1 L asserted, none of the transceivers are enabled, and data is being input from the Unibus on lines BUS D00–BUS D07.

Multiplexer E1 controls passage of status and control information (RXCS) in the form of signals DONE H, INIT ENB (1) H, and TRANSFER REQUEST H from the sequence and function control logic or passage of data from the read/write buffer register (RXDB). If SELECT 02 H is asserted, data is output from E1; if SELECT 02 H is negated, control information (RXCS) is output from E1.

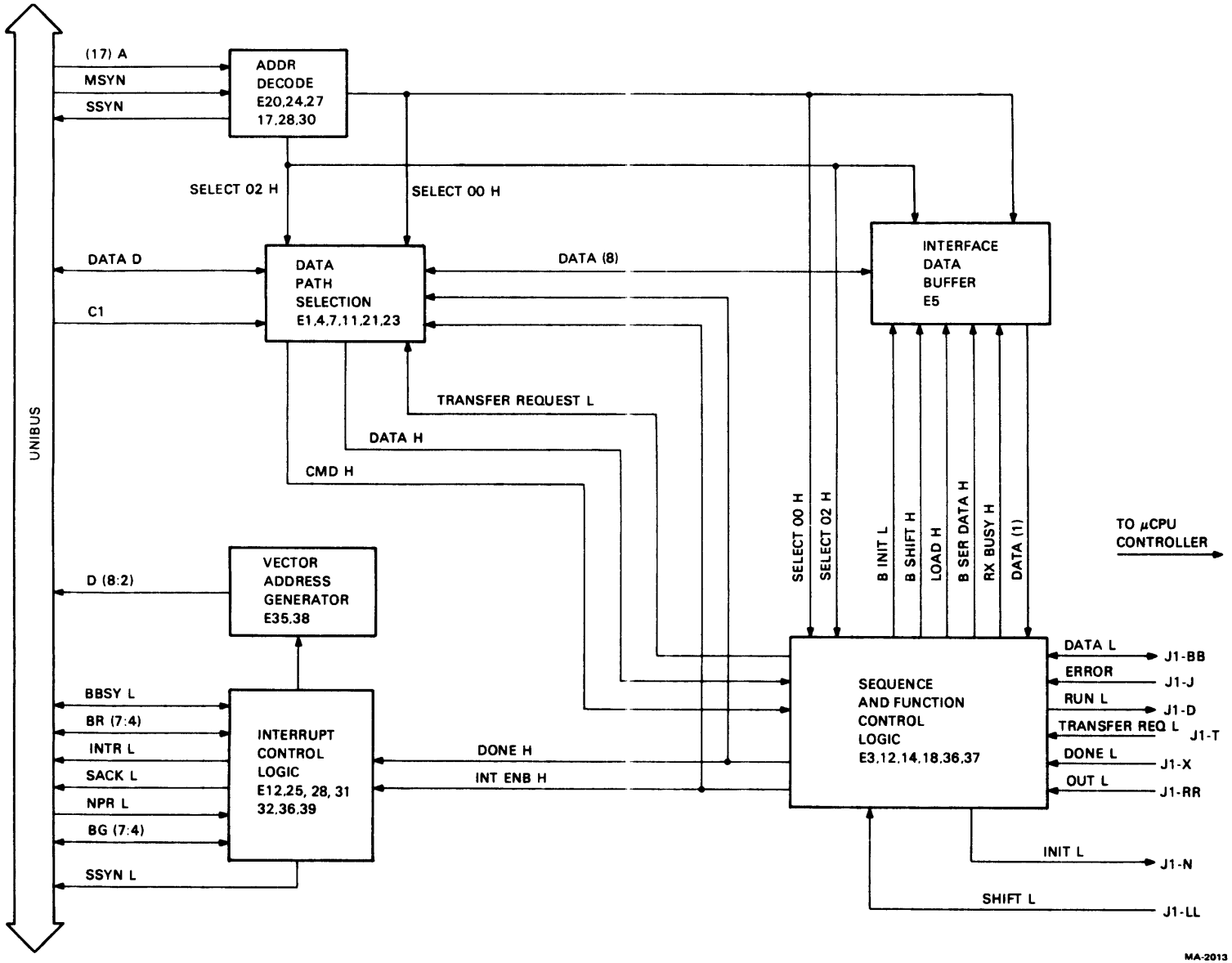


Figure 5-9 RX11 Interface Block Diagram

5.2.2.3 Interface Data Buffer Register – The interface register E5 is an 8-bit, parallel load, shift register. Data transfer through the register can take place from the Unibus to the μCPU controller or from the μCPU controller to the Unibus.

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

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

5.2.2.4 Sequence and Function Control Logic – The sequence and function control logic provides key signals to control the interface register and the interrupt control logic as shown in Figure 5-9. Operation of this circuitry is controlled by signals from the μCPU controller and SELECT 00 H and SELECT 02 H from the address decoder.

Signals RX TRANSFER REQUEST L, RX OUT L, RX DONE L, AND RX RUN L control data transfer between the interface and the μCPU controller. The RX RUN L signal initiates communication between the RX11 interface and the μCPU controller. The run flip-flop E37 is set in passing either a command from interface to controller or data between interface and controller. RUN asserted while Done is true passes a command from interface to controller. Run asserted while Done is false signals transfer of data to or from the controller.

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

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

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

The three signals DONE H, INT ENB (1) H, and TRANSFER REQUEST H from the sequence and function control logic to the data path selection logic represent the three bits that may be read in the control and status (RXCS) register. A functional programming description of this register is given in Paragraph 4.2.

During serial data transfer from the RXDB, binary counter E3 and flip-flop E2 are used to count eight bits of data and to append the parity bit to the data element.

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

5.2.2.5 Interrupt Control Logic – The interrupt control logic receives and generates the control signals required for the RX11 to become bus master. With signals DONE H and INT ENB H both asserted, BUS REQUEST L will be generated if the SACK and BBSY flip-flops, E3, are not set. The BUS REQUEST L signal is routed to the appropriate bus request line (normally BR5) through the priority plug. Etch on the plug selects both request and grant lines. When the bus grant signal is generated by the processor, it is routed via the priority plug and becomes signal BG IN H. This signal clocks the GRANT flip-flop E25 and the SACK flip-flop E31. The SACK flip-flop is set because the RX11 had requested bus mastership. The SACK flip-flop is cleared and the BBSY flip-flop set when BUS BBSY L, BUS SSYN L, and BG IN H are negated on the Unibus. Thus the BUS BBSY L signal will be asserted again by the RX11, which is now bus master. The BBSY signal is inverted and applied to the vector address generator, generating the BUS INTR L signal and the vector address of 264.

5.2.2.6 Vector Address Generator – The vector address generator consists of eight bus drivers E35, E38 that are used to generate a vector address and the BUS INTR L signals. When BUS BBSY L is asserted by the RX11, the inputs to the bus drivers are active. Seven drivers are connected to the Unibus data lines D (08:02) via jumpers. The placement of these jumpers determines the vector address.

5.2.3 RXV11 Interface (M7946) Block Diagram Description

Command and data transfers between the LSI-11 processor and the RXV11 are executed under program control via this module. Figure 5-10 is a block diagram illustrating the logic functions of the interface. The E references on the diagram are IC chip designations on the the RXV11 print set which is a separate document.

5.2.3.1 Address Decoding Logic – Address decoding occurs on the leading edge of BSYNC L assertion. SYNC H clocks address decoding logic to produce an active or passive ME H signal. The ME H signal is a result of comparing DAL REC2–12 H bits to the address configured on address jumpers W7–W17 at SYNC H time. When the RXV11's address is decoded, ME H goes active, enabling an RXV11/LSI-11 bus data or command transfer. Note that address bit DAL REC 1 H is applied to I/O control logic; this bit is used in selecting either command/status (CS) or data buffer (DB) data transfers.

5.2.3.2 I/O Control Logic – I/O control logic circuits control the actual command, status, or read/write data transfer between the LSI-11 Bus and the addressed RXV11 register. Control signals CS H and DB H are generated by this logic function in response to address bit DAL REC 1 H to select either the RXCS or RXDB register. The actual signal sequence for LSI-11 Bus cycle operations involving this function are as described in the *Microcomputer Handbook*.

5.2.3.3 RX Data Buffer (RXDB) Register – The RXDB (E25) is the main command/data interface function. It is an 8-bit parallel load, parallel read shift register. Parallel load occurs during DATO bus cycle execution; RX BUSY H loads command or write data bits present on DAL REC 0–7 H into the shift register. B SHIFT L pulses then serially shift the command or data byte bits out of RXDB bit D07 L and into the serial bus interface and parity logic. Serial command/data and parity bits are then shifted to the controller via the bidirectional RX DATA L signal.

During a data read operation, the process is reversed. Controller serial data bits are received via the RX DATA L signal, serial bus interface and parity logic, and shifted into the RXDB via SER DATA H. Once the data byte is available, a DATI bus cycle can be initiated. Parallel read data bits D0–7 L are gated through input data select logic and routed over TDAL 0–7 H to bus transceivers which place the read data onto BDAL 0–7 L.

5.2.3.4 RX Command/Status (RXCS) Register – The RXCS function is actually not a register. It is a group of command/status bits which are program-accessible via a register address. Only 10 of the 16 RXCS bits are used. Six are write-only bits, three are read-only bits, and one is a read/write bit.

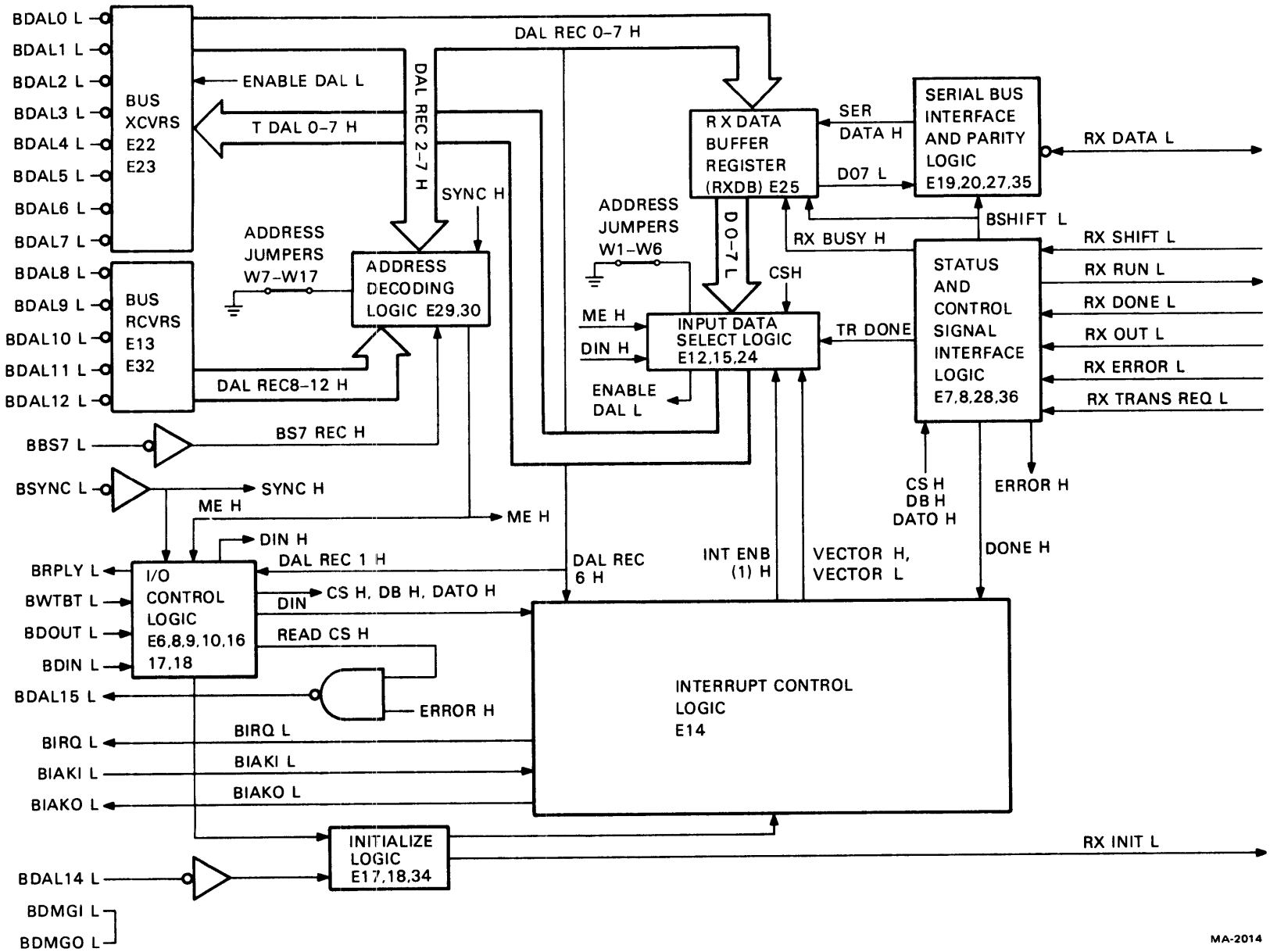


Figure 5-10 RXV11 Interface Block Diagram

During an RXCS read operation, CS H and READ CS H go high, enabling the status of ERROR H, TR, INT ENB (1) H, and DONE bits onto BDAL 15 L, BDAL 7 L, BDAL 6 L, and BDAL 5 L, respectively. Note that input data select logic routes bits 5, 6, and 7 as during the RXDB read operation.

During an RXCS write operation either a command (contained in RXCS bits 1-3) is being transmitted to the RX02, interrupts are being enabled or disabled (RXCS bit 6 set or reset), or the RXV11 is being initialized (RXCS bit 14 is set). RXCS bit 0 (Go bit) is a logical 1 during an RXCS write operation when a command transfer is being executed. This causes the RXDB to parallel-load the command byte and serially transmit it to the controller, as previously described for a data write operation. When RXCS bit 0 is a logical 0 during an RXCS write operation, the command bits are not transmitted to the controller. Instead, the RXV11 is either being initialized (RXCS bit 14 = 1) or the INT ENB (RXCS bit 6) bit is being set or reset.

5.2.3.5 Status and Control Signal Interface Logic - The status and control signal interface logic is the control interface between the controller and the RXV11 interface logic. All control and timing signals required for command or status transfers between the RXV11 interface and the RX02 directly involve this logic function. TR, DONE, and ERROR RXCS signals are produced by this logic function.

5.2.3.6 Interrupt Control Logic - The interrupt control logic function contains the interrupt enable flip-flop (RXCS bit 6). When set, the circuit requests interrupt service when the DONE H signal goes active. The interrupt sequence is initiated by the logic when it asserts BIRQ L. The processor responds by asserting BIAKI L and BDIN L, causing VECTOR H and VECTOR L to go to their respective active states. VECTOR H and VECTOR L cause input data select logic to enable the vector address, configured by jumpers W1-W6, onto the BDAL bus. The actual sequence of operations for interrupt operation is described in the *Microcomputer Handbook*.

5.2.3.7 Initialize Logic - Initialize logic is activated whenever a DATO cycle is executed with the RXCS, and BDAL 14 L is asserted. This is equivalent to writing a logical 1 into RXCS bit 14. The logic responds by generating an active RX INIT L signal (pulse) which initializes the floppy disk drive. RX INIT L only remains active for the duration of the bus cycle.

5.2.4 RX211 Interface (M8256) Block Diagram Description

The RX211 transfers commands and data between the RX02 Controller and the PDP-11. The RX211 is an NPR device which functions as a slave when receiving commands from the PDP-11 and functions as a master when making data transfers. The interface is selected for use by a unique address (normally 177170 for command and status register and 177172 for the data buffer register) which is decoded by the interface so that the command or data can be processed. Figure 5-11 is a block diagram of the RX211 showing the major signal flow to/from the various functional groupings of the module. The E references on the diagram are IC chip designations on the RX211 print set which is a separate document.

5.2.4.1 Address Decoder, Buffer Selector, SSYN Register - When the PDP-11 has a command to be processed by the RX211, address data (BA) and a delayed MSYN signal are placed on the bus. The address bits are decoded by the address Decoder, which by placement of jumpers on E74 (jumper inserted = 0, jumper removed = 1) so that the address and jumper are equal, asserts REG SEL L when MSYN is asserted with the correct address. (Address bits 13:17 are always high.) The REG SEL L along with PROC OUT, which determines direction of transfer, enables the buffer selector circuits to decode BA01, 02 to develop signals which enable the data on the data lines to be entered either into the command and status (C&S) buffer or into the data buffer. When BA01, 02 are both low, data is entered into the C&S buffer, and when BA01 is high and BA02 is low, data is entered into the data buffer. The asserted REG SEL L is also applied to and enables the SSYN register which generates delayed SSYN OUT signals that are applied to the interface control to establish RUN conditions and to the PDP-11 to indicate that the address has been recognized.

5.2.4.2 Command and Status Buffer – Command information on the data lines is entered into the command and status (C&S) buffer circuits when WRT CSR H and SEL CSR H are asserted. The command data is entered in register E37 by WRT CSR H and shifted to the output of multiplexer E14, E8 as MDATA by SEL CSR H. When SEL CSR H is negated, data from the data buffer is output from E14, E8 as MDATA.

5.2.4.3 Data Buffer – The data buffer receives data either from the PDP-11 or from RX02 Controller and couples the particular data to the receiving device via the data input/output circuit. Data from the PDP-11 is parallel loaded into the data buffer E5, E20 when EN SHIFT, developed internally by TR and BDONE, is negated and the clock signal is asserted. The parallel data is shifted serially through the buffer from Data 0 to Data 7 and from Data 8 to Data 15 when EN SHIFT is asserted and the clock signal is asserted. The serial data to be output to the RX02 Controller is coupled to the interface control circuits by bits Data 7, 11, 15. (Data is transferred in 8-bit bytes and commands are transferred in 12-bit format.) SER DATA from the RX02 Controller is serially loaded into the data buffer E5, E20 when EN SHIFT is asserted and shifted by the clock signal. The parallel output of the data buffer is coupled to the data input/output circuits. The buffer clock signal used to shift the data in the buffer is enabled by SHIFT and MSYN CLR EN, RX OUT, WRT REG.

5.2.4.4 Data Input/Output and TRANSMIT DATA CIRCUIT – The data input/output couples the data to be processed between the PDP-11 and the RX02. The inputs to the E1, E2, E4, E7 are C&S data and data buffer data. When data is to be coupled through E1, 2, 3, 7, XMIT DATA L is asserted and the data is available at the output as BD (00:15). For commands, XMIT DATA L is enabled by REG SEL L, and C&S data is applied as MDATA which is subsequently serially shifted through the data buffer to the controller via the interface control. Data buffer data is applied as both MDATA and DATA 1, 2, 10, 12:14 which are parallel loaded on the bus for transfer to the PDP-11. When data is to be applied on the bus from the PDP-11, XMIT DATA L is disabled by RX OUT and PROC OUT. During data transfers, MSYN CLR EN and EN ADDR L are low so that XMIT DATA L is controlled by RX OUT which toggles according to the direction of data transfer from interface to controller or controller to interface.

5.2.4.5 Address Circuits – The address circuits consist of an address counter and address input/output gates which provide the means to transfer data directly to or from PDP-11 memory. For a fill or empty buffer command, the address counter (E61, 55, 49, 32, 26) is loaded with a memory address via the data buffer. For either of these commands, LOAD BAR and WRT REG PLS are asserted and the counter is loaded with BD 01:15. The output of the counter is coupled through the address input/output gates to the bus as BUSA (01:15) which represents the starting address to enter or read from PDP-11 memory. The address counter is incremented after each word (16-bit) data transfer by EN ADDR L which is toggled by the bus control circuit which operates in the NPR mode for the data transfer to fill or empty the controller buffer.

5.2.4.6 Interface Control Circuits – The interface control circuits establish the conditions to transfer commands and data to/from the controller. The function to be performed is identified as fill buffer, empty buffer, read error code, or other by FUNCT 0, 1, 2. When a command is to be transferred, BD00, BDONE, WRT CSR, and SSYN OUT L are used within the interface control (E39, E41) to develop a GO signal whhich asserts RX RUN indicating a command transfer to the controller (while RX DONE is asserted). Data for a command is coupled through the interface control (E9, E15) to the controller via DATA 11 and RX DATA L as a 12-bit word; the SHIFT L signal is used to sequence the interface control. If the function to be performed is fill buffer, empty buffer, or read error code, an NPR REQ is generated to initiate a request for bus mastership. The RX211 establishes bus mastership after data is in the interface buffer. Data is transferred between the controller and the interface in 8-bit bytes using ENA 8 bit or ENA 16 bit to load the controller data in the data buffer or using DATA 7 or DATA 15 to couple data buffer data to the controller. When RX OUT is negated, the TR signal is asserted when the controller is ready to accept the next element of data; when RX OUT is asserted, the TR signal is asserted when a byte of data is transferred from the controller. The RX DONE signal is

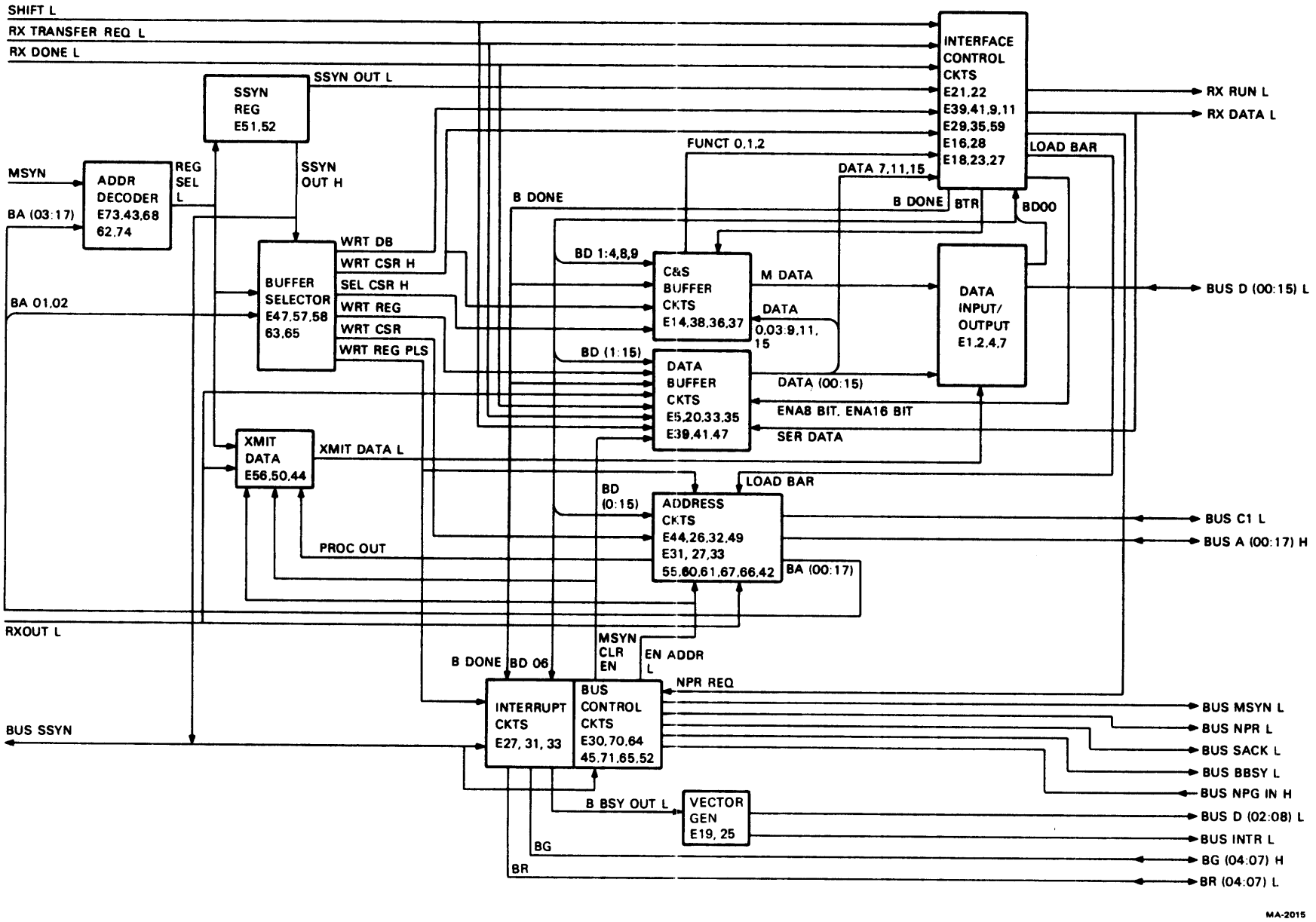


Figure 5-11 RX211 Interface Module Block Diagram

asserted when the function to be performed is completed. After RX DONE is asserted, the 12-bit error status register is transferred; then B DDONE is asserted to establish conditions for performing a new function.

5.2.4.7 Bus Control Circuits – The bus control circuits generate the signals to establish bus mastership along with front-end, tail-end, and data deskew delays required during data transfers. When an NPR REQ is generated by the interface control, a BUS NPR is developed by E30 and applied to the PDP-11 processor. When the bus control circuit receives BUS NPG, it generates an acknowledge, BUS SACK L, a BUS BSY and a delayed (≈200 ns) BUS MSYN indicating it has bus control. EN ADDR L is asserted to enable the memory address to be placed on the bus. SSYN IN is applied from the slave (memory) to indicate conclusioon of data transfer and clears the master Sync (MSYN CLR EN) subsequently negating BUS BSY so another device may take over as bus master.

5.2.4.8 Interrupt Circuits – The interrupt circuits are used to generate an interrupt to return to execution of the PDP-11 program after an RX02 function is completed. When an RX02 function is completed, BDONE iss asserted and if BD06 (interupt enable) of the command and status word was set, a Bus Request is generated to gain control of the bus in order to place the vector address on the bus. [Bus Request is routed to BR (04:07) by the priority plug.] A bus grant from the processor is routed by the priority plug as BG, which is used by E31 to generate a BR SACK signal that generates a B BSY OUT L; the vector generator is enabled and the vector address (BD 02:08) and INTR L are output to the processor. After receipt of the vector address data, the processor asserts SSYN and subsequently bus mastership is released.

5.2.5 RXV21 Interface (M8029) Block Diagram Description

The RXV21 interface transfers commands and data between the RX02 Controller and the LSI-11. The RXV21 is an NPR device which functions as a slave when receiving commands from the LSI-11 and functions as a master when making data transfers. The interface is selected for use by a unique address (normally 177170 for the command and status register and 177172 for the data buffer register) which is decoded by the interface so that a command or data can be processed. Figure 5-12 is a block diagram of the RXV21 showing the major signal flow to/from the various functional grouping of the module. The E references on the diagram are IC chip designations on the RXV21 print set which is a separate document.

5.2.5.1 Input/Output Transceiver, Buffer Selector – When the LSI-11 has a command to be processed by the RXV21, address data BDAL and a BSYNC signal are placed on the bus. The address bits are decoded by the input/output transceiver, which by placement of jumpers (W1 to W11 – jumper inserted = 1, jumper removed = 0) so that the jumper and address are equal, asserts MATCH when BSYNC is asserted with the correct address. MATCH along with BSYNC, which determines direction of transfer, enables the buffer selector to decode TSBUS1 and to develop signals which enable data to be entered either into the C&S buffer or into the data buffer. When TS BUS 1 is negated, data is entered into the C&S buffer (WRT CSR PLS is asserted), and when TS Bus 1 is asserted data is entered into the data buffer (WRT DB asserted). The asserted WRT CSR and WRT DB signals are also applied to the interface control to be used to establish RUN conditions.

5.2.5.2 Command and Status Buffer – Command information on the data/address lines is entered into the command and status (C&S) buffer circuits (E43) when WRT CSR PLS is asserted. Command data is shifted to the output of E38, E39 as TS Bus data by RD CSR L which is asserted when the command is placed on the input lines. The function to be performed is contained in TS BUS 1, 2, 3 which are coupled to the interface control as FUNCT 0, 1, 2 in order to establish the conditions to subsequently shift the 12-bit command to the controller.

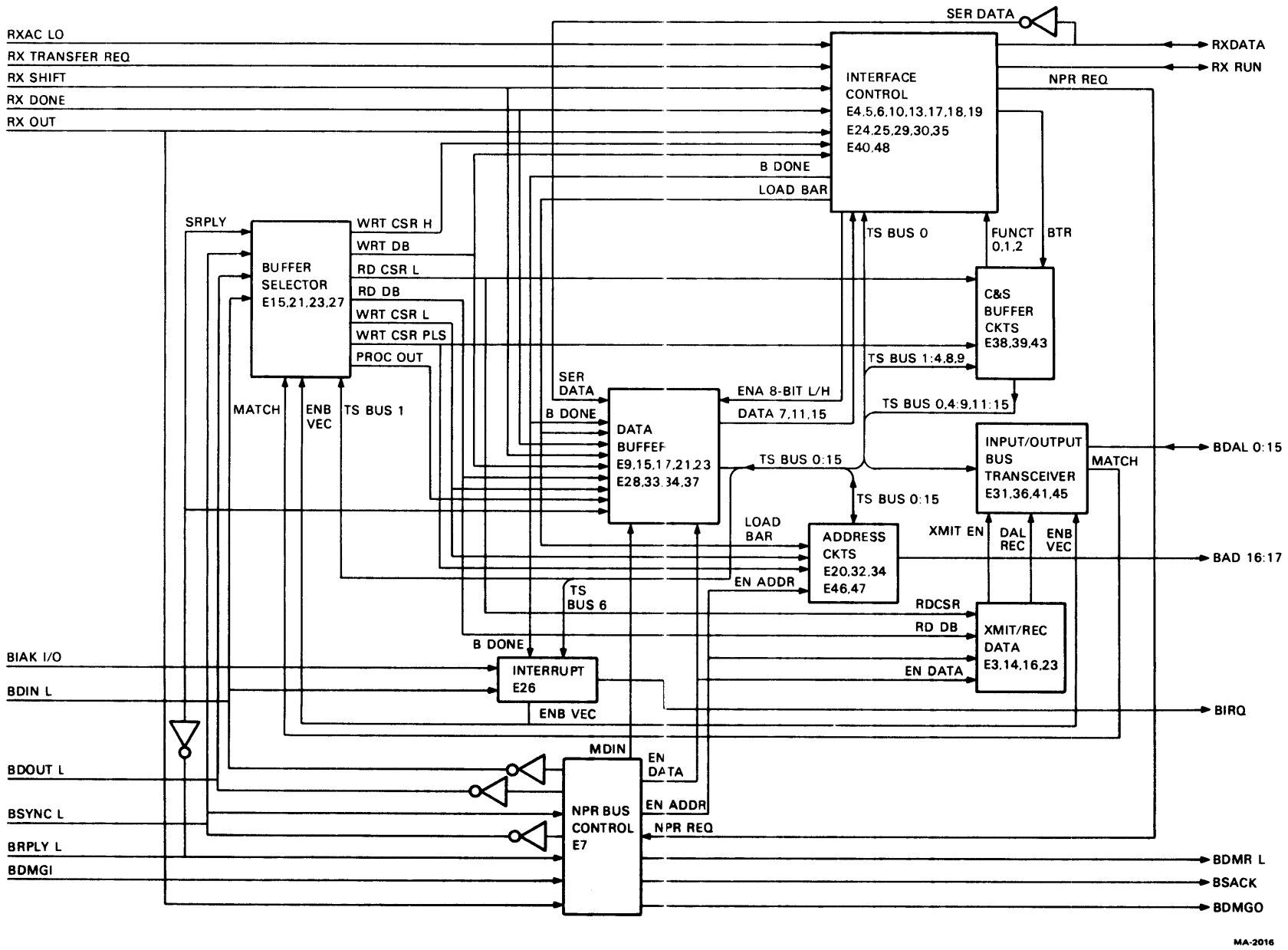


Figure 5-12 RXV21 Interface Module Block Diagram

5.2.5.3 Data Buffer – The data buffer receives data from either the LSI-11 or the controller and couples the particular data to the receiving device via the input/output transceiver. Data from the LSI-11 is parallel-loaded into the data buffer E37, E33 when signals (LO SEL 0 and DB SEL 1), developed internally by WRT DB, WRT CSR, RX DONE, are both asserted and the clock signal developed by SHIFT input is asserted. The parallel data is shifted serially through the buffer from Data 0 to Data 7 and from Data 8 to Data 15 when SHIFT is asserted. The serial data, which is to be output to the controller, is coupled to the interface control circuits by bits Data 7, 11, 15; data is transferred in 8-bit bytes and commands are transferred in 12-bit format. SER DATA from the controller is serial-loaded into the data buffer using ENA 8-bit L and H to enable the high order and low order byte and shifting the data by the clock signal. (The buffer clock signal is enabled by SHIFT and PROC OUT and SRPLY, or LOAD BAR or MDIN.) When 8 bits of data are in the buffer its parallel output is coupled through the input/output circuits to the LSI-11.

5.2.5.4 Input/Output Transceiver and Transmit/Receive Data Circuit – The input/output transceiver couples the data to be processed between the LSI-11 and the controller. Inputs to E31, E36, E41, E45 are C&S data or data buffer data and LSI-11 data/address. When data is to be coupled from the LSI-11 through the input/output, DAL REC is asserted and the data is available as TS BUS 00:15. The DAL REC signal is enabled by negating RD DB, RD CSR, EN DATA and EN ADDR. The TS BUS data is applied to the C&S buffer and the data buffer and all data (commands and data) are subsequently serially shifted through the data buffer to the controller via the interface control. When data is to be coupled to the LSI-11, XMIT EN is asserted and data buffer data or status data are coupled through the input/output transceiver to the bus for transfer to the LSI-11.

The XMIT EN signal is asserted by negating RD DB or RD CSR or by asserting EN ADDR or EN DATA. Status data are output when RD CSR is applied to the C&S buffer circuits and data from the data buffer are output when RD DB is applied to the data buffer. The direction of data transfer from the interface to the controller or from the controller to the interface is determined within the buffer selector circuits according to the state of BD IN or BD OUT.

5.2.5.5 Address Circuits – The address circuits consist of a counter which provides the means to directly address the LSI-11 memory for data transfers. For a fill or empty buffer command, the address counter (E46, E47) is loaded with a memory address via the data buffer. For either of these commands, LOAD BAR is asserted and the counter is loaded from TS Bus 00:15. (For an extended address, TS Bus 12, 13 are loaded into E 32 when the counter reaches maximum by WRT CSR and WRT CSR PLS during a command word input so that bits BAD 16, 17 are available.) The output of the counter is coupled through the input/output transceiver to the bus as BDAL 0:15 which represents the starting address to enter or read from the LSI-11 memory. The address counter is incremented after each word (16 bits) transfer by EN ADDR L which is toggled by the bus control circuit operating in the NPR mode for the data transfer to fill or empty the controller sector buffer.

5.2.5.6 Interface Control Circuits – The interface control circuits establish the conditions to transfer commands and data to/from the controller. The function to be performed is identified as fill buffer, empty buffer, read error code, or other by FUNCT 0, 1, 2. When a command is to be transferred, TS Bus 0, B DONE, WRT CSR are used within the interface control (E17) to develop a Go signal which asserts RX RUN indicating a command transfer to the controller while RX DONE is asserted. The command word is coupled through the interface control (E13, E48, E35) to the controller via Data 11 and RX DATA L as a 12-bit word; the RX SHIFT signal from the controller is used to step the interface control through its sequence. If the function to be performed is fill buffer, empty buffer, or read error code, an NPR REQ is generated to initiate a request for bus mastership. (If RX AC LO is asserted indicating a power loss in the controller, NPR REQ is inhibited.) The RXV21 establishes bus mastership after data is in the interface buffer. Data is transferred between the controller and the interface in 8-bit bytes using ENA 8-bit L and H to load the controller SER DATA in the data buffer or using DATA 7 or DATA 15 to couple data buffer data to the controller. The RX OUT signal indicates the direction of transfer. When RX OUT is negated, the TR signal is asserted when the

controller is ready to accept the next element of data, and when RX OUT is asserted, the TR signal is asserted when a byte of data is transferred from the controller. The RX DONE signal is asserted when the function to be performed is completed. After RX DONE is asserted, 12-bit error status is transferred, then B DONE is asserted to establish conditions for performing a new function.

5.2.5.7 Bus Control Circuits – The bus control circuits generate the signals to establish bus mastership during data transfers. When an NPR REQ is generated by the interface control, a BMDRL signal is developed by E7 and applied to the LSI-11 processor. A bus grant (BMGI) is applied to the bus control circuitry (BRPLY and BSYNC are negated) to grant bus mastership and an acknowledge (BSACK L asserted) is generated indicating the interface has bus mastership. Signal EN ADDR L is asserted to enable memory address to be placed on the bus. After becoming master, B SYNC L and either BD IN L or BD OUT L are asserted, data is placed on the Bus, BRPLY is received, the BD IN or BD OUT is negated, B SYNC is negated, and bus mastership is released.

5.2.5.8 Interrupt Circuits – The interrupt circuits are used to generate an interrupt to return to the execution of the LSI-11 program after an RX02 function is completed. When an RX02 function is completed, B DONE is asserted and if TS BUS 6 (interrupt enable) of the command and status word was set, a BIRQ (bus request) is generated to gain control of the bus in order to place the vectoraddress on the bus. A bus grant (BIAKI) from the processor is routed by the priority daisy-chain and is used by E26 to generate ENB VEC which enables the vector address (BDAL 02:08) from the input/output transceiver and BRPLY (via the buffer selector). After receipt of the vector address, the processor terminates BD IN and BIAKO and subsequently bus mastership is released.

5.3 UNIT LEVEL DESCRIPTION

The following paragraphs provide a functional description of the units housed within the RX02 cabinet. There is a block diagram description of the μCPU controller, the read/write electronics, and the mechanical drive; there is also a brief description of the μCPU software. These units contain all the components necessary to rotate the disk, position the read/write heads, and perform the desired functions (write, read data). (Signal flow and interconnection between these units and the interface module are described in Paragraph 5.1.)

5.3.1 Microprogrammed Controller (M7744) Hardware Description

This section describes the hardware operation of the controller. (The μCPU software is described in Paragraph 5.3.2.) Figure 5-13 is a detailed block diagram of the controller showing the major signal flow to/from the various functional groupings that develop the signals to accomplish the various RX02 functions (e.g., write data, read data, etc.). The E references on the diagram are IC chip designations on the controller print set which is a separate document.

5.3.1.1 PROM, ROM Register, Processor and Sequencer Circuits – Operation of the controller is controlled by the PROM, ROM register, microprocessor, and microsequencer circuits which function as a computer control unit. The PROM contains a 1538, 16-bit word microprogram which determines controller operation. The output of the PROM, which is determined by the ROM ADDR (10:0) input from the microsequencer address register, is applied to the ROM register and microprocessor.

The ROM register is used to store the microinstruction and subsequently controls operation of the microprocessor, microsequencer, and the various other circuits which develop the control signals to accomplish the desired microinstruction function. The microprocessor performs arithmetic and logical operations specified in the 16-bit microinstructions of program and provides an 8-bit data and status output (μDAT 0:7) that contains eight branch conditions for the branch selector.

There are seven types of microinstructions contained in the microprogram: arithmetical and logical, branch (condition high and condition low), I/O (input/output), Jump and JSR instructions. The formats for the various instructions output from the ROM register are shown in Figures 5-14, 5-15, 5-16 and 5-17.

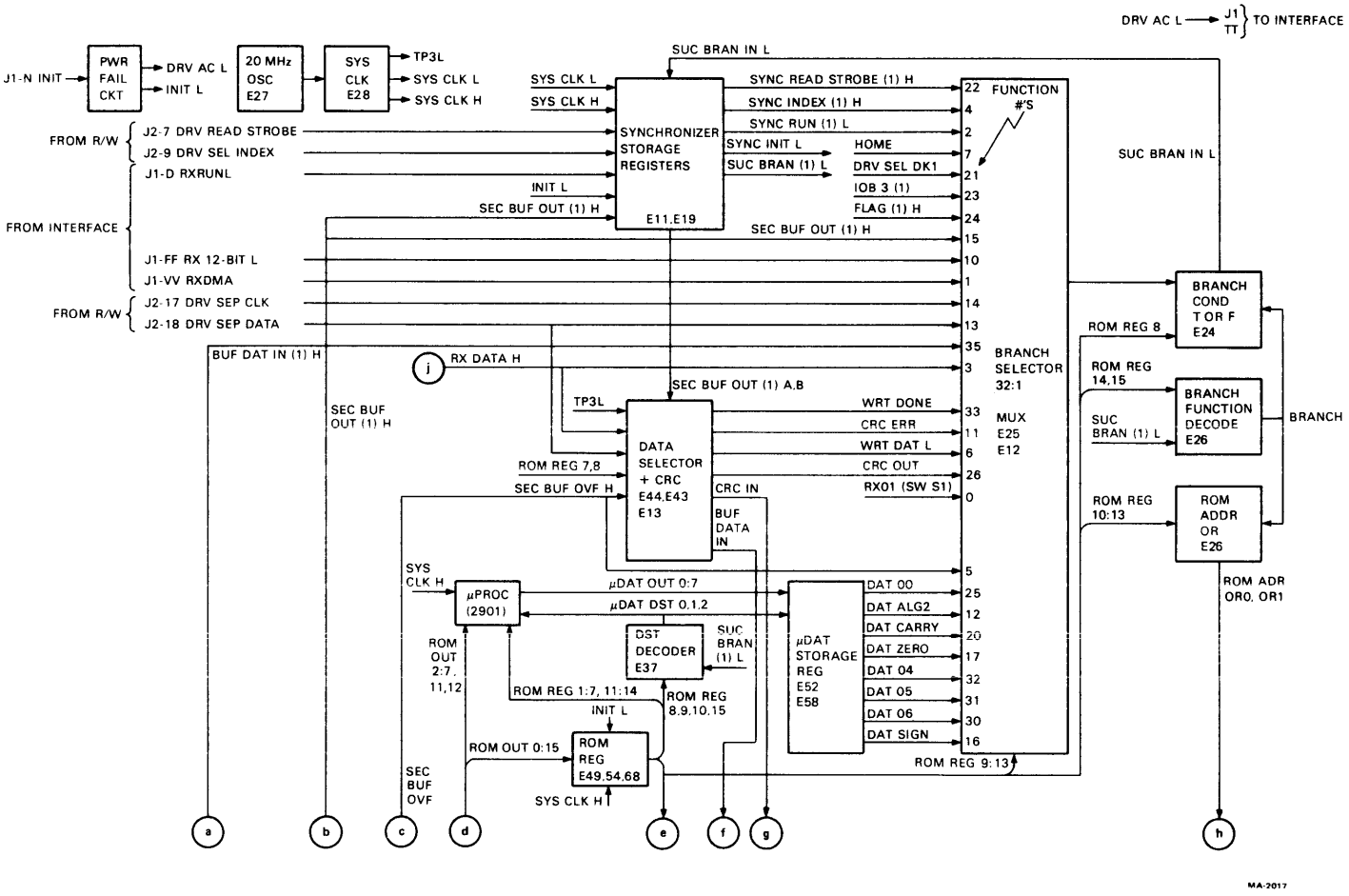


Figure 5-13 μCPU Controller Block Diagram
(Sheet 1 of 2)

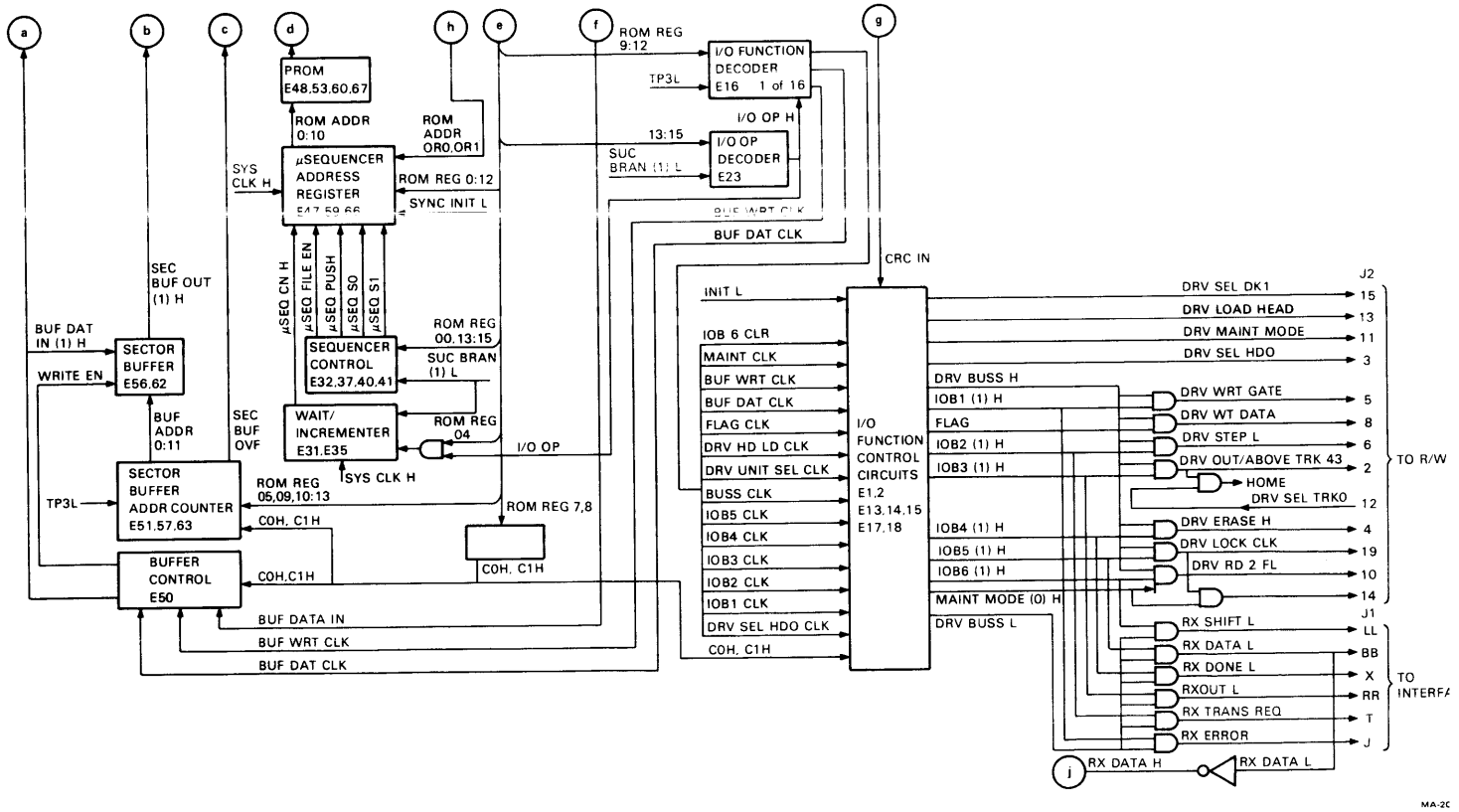


Figure 5-13 μCPU Controller Block Diagram
(Sheet 2 of 2)

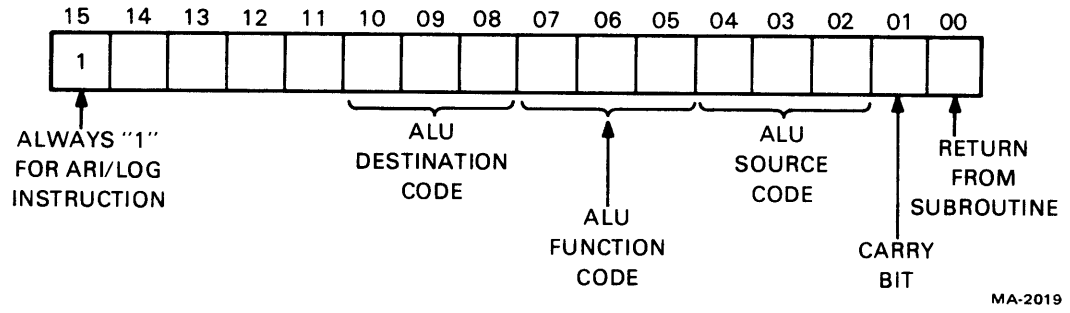


Figure 5-14 Arithmetic and Logical Instruction Format

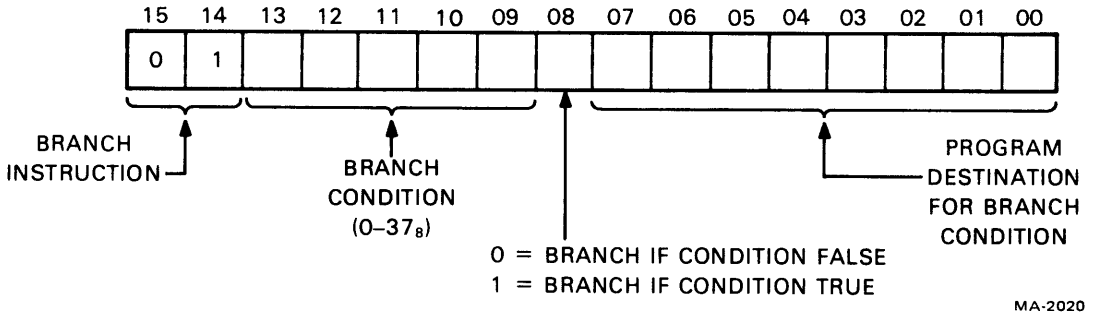


Figure 5-15 Branch Instruction Format

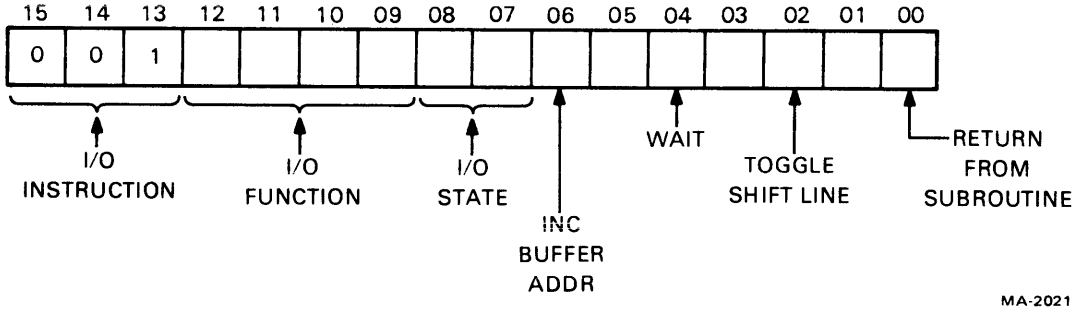


Figure 5-16 I/O Instruction Format

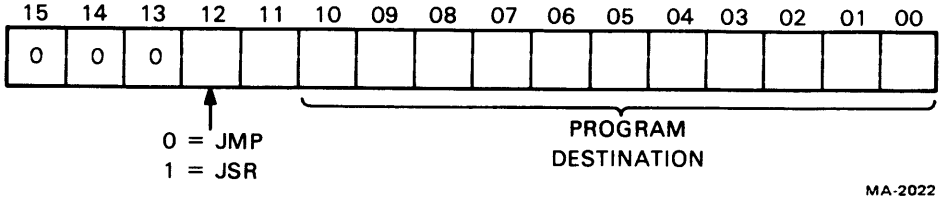


Figure 5-17 JMP/JSR Instruction Format

For an arithmetic and logical instruction the output of ROM REG bit 15 is equal to 1; this signifies that this instruction is to be performed by the microprocessor. ROM REG bits 02, 03, 04 indicate the source of data (ROM REG, ROM OUT, or internal inputs) for the microprocessor; ROM REG bits 11, 12, 13, 14 provide A and B RAM addresses for the processor; ROM REG bits 05, 06, 07 indicate the function to be performed by the microprocessor (add, subtract, etc.); and ROM REG bits 8, 9, 10 are decoded by DST decoder E37 when bit 15 is a 1 to indicate the data destination within the microprocessor and also the output of the microprocessor. The output of the microprocessor is applied to the μDAT storage register E52, E58 which is loaded whenever the destination code μDAT DST 0, 1, 2 is other than 0, 0, 1, respectively; this code indicates no storage (i.e., NO-OP - no change to data registers in 2901) and a constant is input to the microprocessor for the microinstruction immediately before the microinstruction when NO-OP occurs. The constant input to the microprocessor for arithmetic operations is contained in PROM outputs, ROM OUT 2 to 7, 11 and 12 which are used on the microinstruction currently at the output of the ROM register. When the constant is loaded in the ROM register, a NO-OP or "waste" cycle occurs and the microprocessor is unmodified.

For a branch instruction, ROM REG bits 15 and 14 are 0 and 1, respectively; this signifies that this is a branch condition as decoded by branch decoder E26. ROM REG bits 13, 12, 11, 10, and 9 indicate the branch condition which is decoded by the 32-to-1 branch selector E12, E25. ROM REG bit 8 indicates whether a high or low signal will cause a successful branch. An unsuccessful branch takes one cycle and causes the microprocessor program to be incremented by one. A successful branch takes the cycles and branches to the address formed by bits (07:00) concatenated with the microprogram counter three high order bits.

For an I/O instruction, ROM REG bits 15, 14, 13 are 0, 0, and 1, respectively; these bits are decoded by I/O OP decoder E23. ROM REG bits 12, 11, 10 and 9 indicate the I/O function to be performed which is detected by I/O function decoder E16. ROM REG bits 8 and 7 are decoded as C0 and C1 which determines the state (setting, resetting, or toggling a flip-flop) of I/O function control circuits E1, 2, 13, 14, 15, 17, and 18.

For JMP and JSR instructions, ROM REG bits 15, 14, 13, and 12 are 0000 (JMP) and 0001 (JSR) respectively; these bits are decoded by the microsequencer control E31, 37, 40 and 41. ROM REG bits 10–1 contain the microprogram address of the jump.

As each microinstruction is output from the PROM, the next instruction is determined by the microsequencer address register. The output of the microsequencer register is governed by the microsequencer control (E31, 37, 40, 41) and the wait/incrementer (E34, E35). The microsequencer control decodes ROM REG bits 00, 13:15 and generates the μSEQ FILE EN, μSEQ PUSH, μSEQ S0, and μSEQ S1 signals that determine the microsequencer internal source for the next address. The wait/incrementer and decoding of the micro op code generates the μSEQ CN H signal which is used either to increment the microsequencer register to the next sequential address or to hold the microsequencer register to the current address, in which case the same microinstruction is repeated until the wait/incrementer overflows or a successful branch occurs. After an arithmetic and logic instruction, the next PROM address is from the microsequencer internal program counter which enables the program to continue in sequence. After a successful branch instruction, the nexxt PROM address is from the microsequencer program counter bits (10:8) concatenated with the microsequencer D inputs (ROM REG bits 07:00) which enables the microprogram to branch to an address other than the next sequential address. (For a write double density branch, the ROM ADDR 0R0, 0R1 signals are ORed into the least significant bits of the microinstruction and a normal successful branch is never performed.) For an unsuccessful branch, the microprogram counter is incremented. After an I/O instruction, the next PROM address is from the microsequencer internal program counter. When the wait bit (ROM REG bit 4) is set in an I/O instruction immediately before a branch instruction, the program is incremented by μSEQ CN H after a wait and a successful branch occurs; if the branch is unsuccessful, the program is incremented after the wait elapses. (The incrementer countt is set to zero and when a

successful branch occurs, the count is preset to 256 and the μSEQ CN H goes high.) For either JMP or JSR instructions, the program jumps unconditionally to the PROM address which is contained in the current ROM REG (10:00) inputs applied to the R inputs of the microsequencer register; the program returns to the next instruction after the JSR instruction via the address stored in the microsequencer internal file.

5.3.1.2 Branch Control Circuits - The branch control circuits, which cconsist of the branch selector (E12, E25), the Branch Cond High or Low (E24), the branch function decoder (E26), and the ROM ADDR OR gate (E26), detect when a branch condition is to be performed and develop a low SUC BRAN IN L signal to indicate a successful branch. There are 32 branch conditions that are used for the microprogram. When a branch instruction is at the output of the ROM register, a low ROM REG bit 13 enables the branch selector to decode ROM REG bits 12, 11, 10, and 9 which contain the code for the particular branch. (MUX E12 is used for branches $0-17_8$ and MUX E25 is used for branches 20_8-37_8.) The output of the branch selector, which is the inverse of the input, is applied to branch condition T or F gate which is enabled by the BRANCH signal (ROM REG 15, 14 = 0, 1) to develop a low SUC BRAN IN L when some branch conditions are false (ROM REG 08 = 0) and when some conditions are true (ROM REG 08 = 1). The low SUC BRAN IN L signal, which indicates a successful branch, forces the BRANCH signal false and after being applied to the synchronizer storage register, enables the microsequencer to increment to the next microinstruction address. The ROM ADDR OR gate provides the ROM ADDR 0R0, 0R1 to the microsequencer to modify the PROM address during double density writing (ROM REG bits 13, 12, 11, 10 = 1111) while a branch condition is incomplete; this only happens for branch conditions 36_8 and 37_8.

If it is assumed that the ROM REG output contains a branch instruction (01 00110 000 110 000) to test for write data, the following events would occur. ROM REG bits 15 and 14 are 0 and 1, respectively, so the BRANCH signal at E26 is high. ROM REG bits 13, 12, 11, 10, 9, are 0, 0, 1, 1, 0 respectively so that MUX E12 is enabled by ROM REG bit 13 low and the WRT DAT L input to the MUX is selected to be output from the MUX E12. ROM REG bit 8 is 0 so a branch will occur when the write data condition (WRT DAT L) is false. So for this instruction, when write data is not in process (WRT DAT L = 1), a branch will occur (SUC BRAN IN L = 0) and the microprogram will go to instruction 060. (ROM REG bits 7, 6, 5, 4, 3, 2, 1, 0 are 00110000 respectively.)

5.3.1.3 I/O Control Circuits - The I/O control circuits which consist of the I/O OP decoder (E23), the I/O function decoder (E16) and the I/O function control (E1, 2, 13, 14, 15, 17, 18), detect when an I/O (input/output) function is to be performed and develop the appropriate signals (RX DATA L, RX OUT L, etc.) to accomplish the desired results (read data, write data, etc.). There are 16 I/O functions that are to be accomplished by the microprogram. When an I/O instruction is at the outputof the ROM register, ROM REG bits 15, 14, 13 (001) are decoded by the I/O OP decoder to enable the I/O function decoder to decode ROM REG bits 12, 11, 10, 9 which contain the code for the particular I/O function. The output of the decoder is 1 of 16 clock signals, operating at the rate of TP3 L (50 ns pulse every 200 ns), which are applied to the I/O function control circuits, the CRC, and the buffer control; these circuits develop interface signals between the controller and the R/W electronics and the interface modules. The I/O clock signals trigger the control circuits to set, clear, etc. according to the enabling signals C0, C1 which are generated by ROM REG bits 8 and 7 (00-Toggle, 01-set, 10-clear, 11-no op). The clock signals (BUF DAT CLK and BUF WRT CLK) applied to the buffer control are used to write/read buffer data also according to the enabling signals C0, C1. As each I/O control flip-flop is triggered, its output is applied to selection gates where, according to the bus enabled (DRV BUSS H = R/W, DRV BUSS L = interface), the desired signal is available at the controller output.

If it is assumed that the ROM REG output contains an I/O instruction (001 000 1010 000 000) to set Write Gate, the following events would occur. ROM REG bits 15, 14, and 13 are 001 respectively so the I/O OP H signal at E23 is high. ROM REG bits 12, 11, 10, 9 are 0001, so that when TP3 L goes low, the IOB 1 CLK L signal goes low at the output of E16. The IOB 1 CLK L siignal is applied to and sets E13 making IOB 1 (1) H high because ROM REG bits 8 and 7 are 0 and 1 respectively. With IOB (1) H high, it is ANDed with a DRV BUSS H signal, which would have been generated prior in the microprogram instructions, so DRV WT GATE H signal is applied to the R/W electronics to enable the write drive.

5.3.1.4 Sector Buffer and Control Circuits – The sector buffer circuits which consist of buffer control (E50), sector buffer address counter (E51, 57, 63) and the sector buffer (E56, E62) stores a sector of data either input from the interface module or input from the R/W electronics. The stored data from the interface module is subsequently written on the disk and the stored data from the R/W electronics (read from the disk) is subsequently output to the interface module. Whether the sector buffer stores or outputs data is determined by the WRITE ENABLE output of buffer control E50. When WRITE ENABLE is low, data can be stored in the buffer (C1 H asserted and BUF WRT CLK L asserted) and when WRITE ENABLE is high, data can be output from the buffer (C0 H asserted and BUF WRT CLK L asserted). The data is input to the sector buffer serially via the buffer control (BUF DATA IN H) which is applied from the data selector and CRC circuit. The BUF DATA IN H signal is applied to flip-flop E50 which is triggered by BUF DAT CLK L once for each bit; E50 is set or cleared according to the data to be entered, and the BUF DAT IN (1) H output of E50 is input to the sector buffer and also to the branch selector where the program tests for a write data condition. The data is output serially from the sector buffer as SEC BUF OUT (1) H when WRITE ENABLE is negated, and the buffer is either emptied or filled according to the BUF ADDR output of the buffer address counter.

The buffer address counter can be preset to count either 1024 (single density data) or 2048 (double density data) and then it outputs a SEC BUF OVF (overflow) to the data selector and branch selector circuits when an overflow occurs. The counter is preset when ROM REG bit 13 is 1 (I/O function) and ROM REG bits 9 to 12 contain the I/O function code 16; the counter is preset to count 1024 when C0 H is low and C1 H is high and it is preset to count 2048 when both C0 H and C1 H are low. Once the counter is preset, it is incremented by TP3 L whenever ROM REG bit 5 is asserted during an I/O function. As the counter is incremented, a new address is applied to the sector buffer in order to either fill the buffer or empty the buffer; the buffer address is always sequential up to 1023 or 2047.

5.3.1.5 Data Selection and CRC Circuits – The data selection and CRC circuits, which consist of a 4-to-1 MUX E44 and 16-bit shifter E43, controls what data is input to the sector buffer, when data is output from the sector buffer, and also generates a CRC character to be written on the disk and checks the CRC character read from the disk. When data is to be input to the sector buffer, RX DATA H from the interface module is selected at E44 if ROM REG bits 7 and 8 are 1 and 0 respectively, and DRV SEP DATA H from the R/W electronics is selected if ROM REG bits 7 and 8 are both 0. The BUF DAT IN H serial output of E44 is the same as the input ("1" in "1" out, "0" in "0" out). When data stored in the sector buffer is to be output, SEC BUF OUT B is selected as the input at E44 when ROM REG bits 7 and 8 are 0 and 1 respectively, and the output CRC IN is applied to the I/O function control circuits for subsequent data output to the interface or R/W electronics.

The CRC character is calculated when data is being written on the disk by applying the CRC IN to CRC generator E43 at the same time it is being used as an output to the driver in the R/W electronics. The CRC generator E43 is preset, to implement the $2_{16} + 2_{12} + 2_5 + 1$ polynomial, by a low DRV BUSS H when data is to be input from the interface. The CRC E43 is enabled by a high input from flip-flop E13 and data is clocked into the CRC by TP3 L. The CRC generator remains enabled until SEC BUF OVF H occurs at which time CRC OUT occurs and the CRC character (16 bits) is written. After the data field has been written, the CRC register contains the remainder of the division by the polynomial, which is a two-byte character that is written after the data field.

While the data is being output from the buffer and while the CRC character is being output, WRT DAT L is low to indicate that writing a sector of data is still in progress. After the CRC character is written, WRT DONE L goes low and the microprogram branches to another instruction. The CRC character is checked when data is being read from the disk by using DRV SEP DATA to develop CRC INN at E44 at the same time DRV SEP DATA is being used to develop BUF DAT IN. The CRC IN is applied to CRC E43 which checks all bits including the CRC bits. The data and CRC bits are divided by the same polynomial so that if an error is detected, CRC ERR H goes high; otherwise it remains low.

5.3.1.6 Timing and Synchronizing Circuits – The timing circuits consist of a 20 MHz oscillator E27 and a counter E28. This circuit develops 5 MHz SYS CCLK H AND SYS CLK L signals and a 5 MHz TP3 L signal that has a 50 ns pulse width. These timing signals are used throughout the controller to time the shifting and storing of data.

The synchronizing circuits consist of two flip-flops, E11 and E19. These circuits are used to synchronize external signals and delayed internal signals to coincide with timed events within the controller. Pulsed signals such as DRV SEL INDEX, as well as signals with levels, are clocked into E11 by SYS CLK L and then shifted into E19 by SYS CLK H, thus synchronizing these signals with events of the microsequencer, microprocessor, ROM register, etc. which are all clocked by SYS CLK H.

5.3.1.7 Power Fail Circuit – The power fail circuit is used to clear the controller registers and stop timing when the RX02 is initialized or loses power. When power is first applied, RX INIT L is low and forces both DRV AC L and INIT L low to clear the various registers; then two transistors within the circuit conduct and force both DRV AC L and INIT L high so the various registers are ready to operate. When power is lost in the RX02, these same two transistors stop conducting and DRV AC L and INIT L are both forced low.

5.3.2 Microprogrammed Controller Software Description

The following paragraphs provide a brief description of the various subroutines of the controller microprogram. The flowcharts presented in Figures 5-18 to 5-30 provide a guide through the microprogram listing which is contained in the RX02 print set.

5.3.2.1 Initialize Routine (Figure 5-18) – This subroutine starts at microinstruction 0. When the host processor initializes or there is a power loss in the RX02, the program counter is cleared and the RX02 stops. When the microcode initialize is negated, unit (drive) one is selected and the head is stepped in and out, then unit zero is selected and the head is stepped in and out. If errors are found, error (error code 010 for drive 0 and error code 020 for drive 1) and done are set. If there are no errors, a maintenance mode check is initiated in order to check out the read path of the read/write electronics. If there are no errors in the maintenance check, drive zero is checked to see if it is up to operating speed and the ready condition is flagged. The density is determined and sector 1 of track 1 is loaded into the sector buffer of the controller, the drive status is set into the interface register, and thenn the routine ends.

5.3.2.2 Find Header (FIND HD) Subroutine (Figure 5-19) – This subroutine which starts at microinstruction 400 is used to locate the identification (data, clock) address mark of the sector header field. A search is made for the header ID, and if this is successful, the track and sector are compared with the target track and sector. If there is a mismatch between tracks and sector, the mismatch is flagged. A byte of zeros is read and a check is made for CRCC errors; if there is an error, the header bad start count is incremented and the header search continues. If there is no error but there is a mismatch between the target track and the actual track read, error and done are set (error code 150). If there are no errors, the subroutine returns to the calling routine.

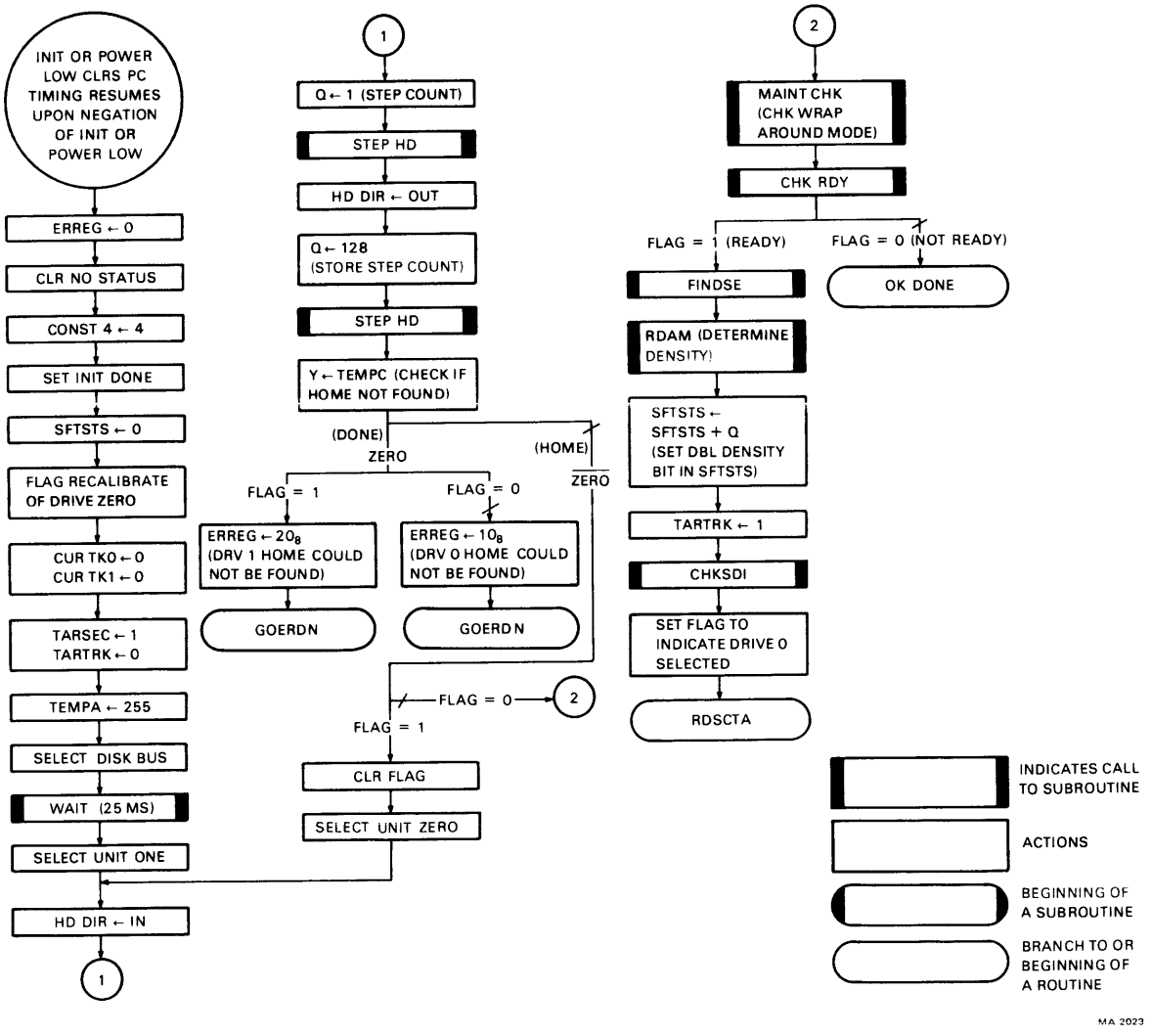


Figure 5-18 Initialize Routine Flowchart

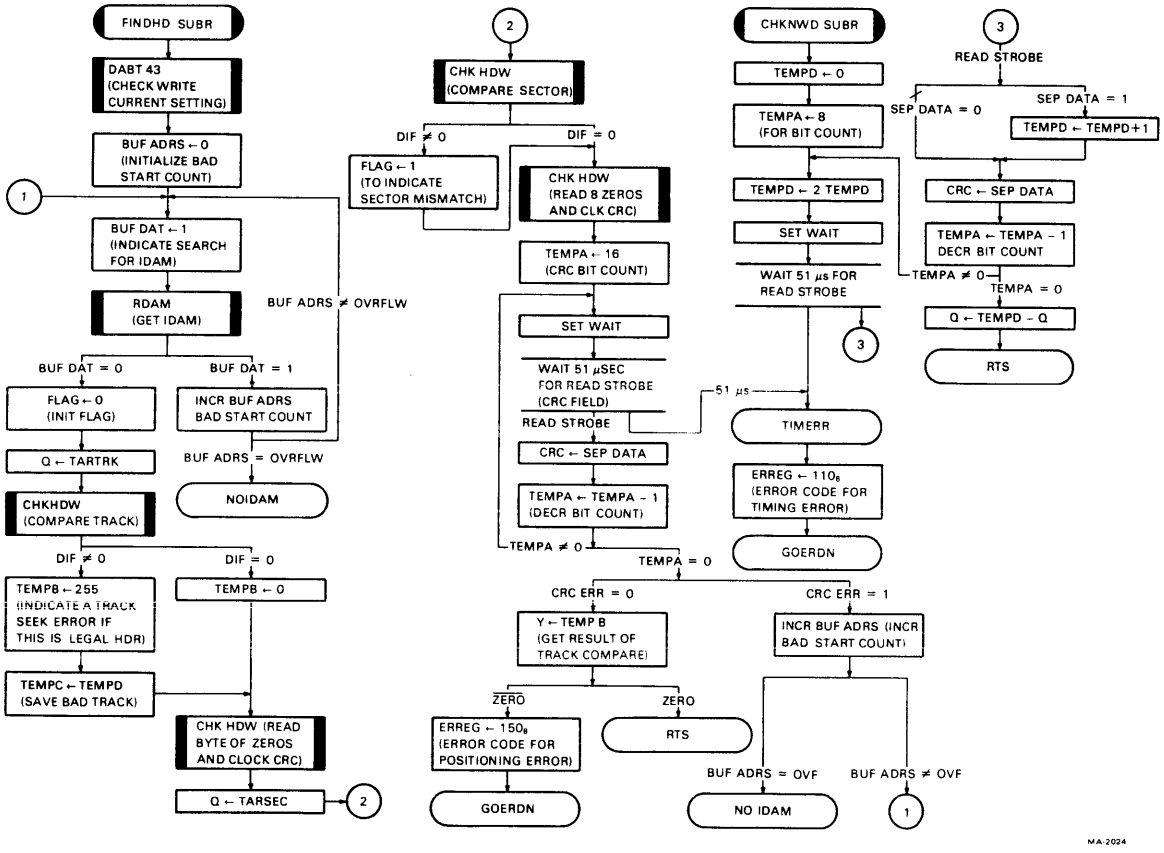


Figure 5-19 Find Header Subroutine Flowchart

5.3.2.3 Read Address Mark (RDAM) Subroutine (Figure 5-20) – This subroutine which starts at microinstruction 471 is used to identify the address mark. A check is made for single or double density, the drive bus is selected, and lock clock is set to enable a read from the drive. There is a 51 μs wait for Read Strobe to occur. If there is no read strobe, an error (error code 120 for nonexistent drive) is set, but if read strobe occurs, a check is made for separated data (a nonzero first bit). If data "one" is not detected before 255 zeros have passed, error is set. Each bit of the header ID byte is checked for the appropriate data and separated data, and when the ID is identified, the subroutine returns to the calling routine. If no IDAM is found, error and done are set (error code 160).

5.3.2.4 Read (RD) Sector Subroutine (Figure 5-21) – This subroutine which starts at microinstruction 1657 is used to read a sector of data into the sector buffer. First the track and sector are located, the data address mark is read, and density is checked. Data is read from the disk, written into the buffer, and applied to the CRC. When the complete sector is written into the buffer, CRC is read and if there is no error, the function is ended (OK DONE). If there is a CRC error, error and done are set (error code 200).

5.3.2.5 Write/Write Sector Subroutine (Figure 5-22) – The write sector and write routines start at microinstructions 363 and 124, respectively, and they are used to write a sector of data on the disk. The write sector subroutine starts by locating the track and sector and then goes to the write subroutine. Data is written on the diskette a single sector at a time. To write data in a sector, the drive bus is selected, the high or low level write current is selected, the density is determined, the data from the sector buffer [either 1024 bits (single density) or 2048 bits (double density)] is written, the CRC (16-bits) is written, and then the postamble is written. The DAM (data address mark) appropriate to either a write or write sector command is written immediately before the data field.

5.3.2.6 Read Error Register (RDERRG) and Set Density (SET DEN) Subroutines (Figure 5-23) – The RDERRG subroutine which starts at microinstruction 1336 is used to apply error status to the interface register. The set density subroutine which starts at microinstruction 1251 is used to set the density (data address mark) for each sector of each track on the diskette.

The set density subroutine starts by filling the sector buffer with zeros and then selecting target track 0 and target sector 1. The appropriate density is written on sector 1 of track 0 and the sector is filled with zeros. The sector count is incremented until all sectors of a track have the appropriate density and all zeros; then the track count is incremented. When all sectors of all tracks are filled with the appropriate density and zeros, the subroutine returns to the calling routine.

The RDERRG subroutine starts by asserting RX OUT to establish that the direction of data transfer is to the interface module. If the operating configuration is the same as an RX01 (no DMA interface), the content of the error register is output to the interface register. But if there is a DMA interface, the word count, current track 0, current track 1, target track, target sector, and soft status registers are applied to the interface and the subroutine exits to OK DONE.

5.3.2.7 Fill/Empty Buffer Routine (Figure 5-24) – This routine which starts at microinstruction 1000 is used to fill or empty the sector buffer with single or double density data. The direction for data transfer is determined and OUT is set accordingly; then it is determined if the interface is DMA or programmed I/O. The appropriate word or byte count is set according to single or double density and 8-bit or 12-bit mode. The buffer address is incremented or decremented as appropriate for the fill or empty commands and then the routine exits to OK DONE. (When filling the buffer for word counts less than maximum, the unused portion of the buffer is filled with zeros.)

5.3.2.8 Find Track Subroutine (Figure 5-25) – This subroutine which starts at microinstruction 1124 is used to locate the target track for a particular drive and then to step the read/write head to that track.

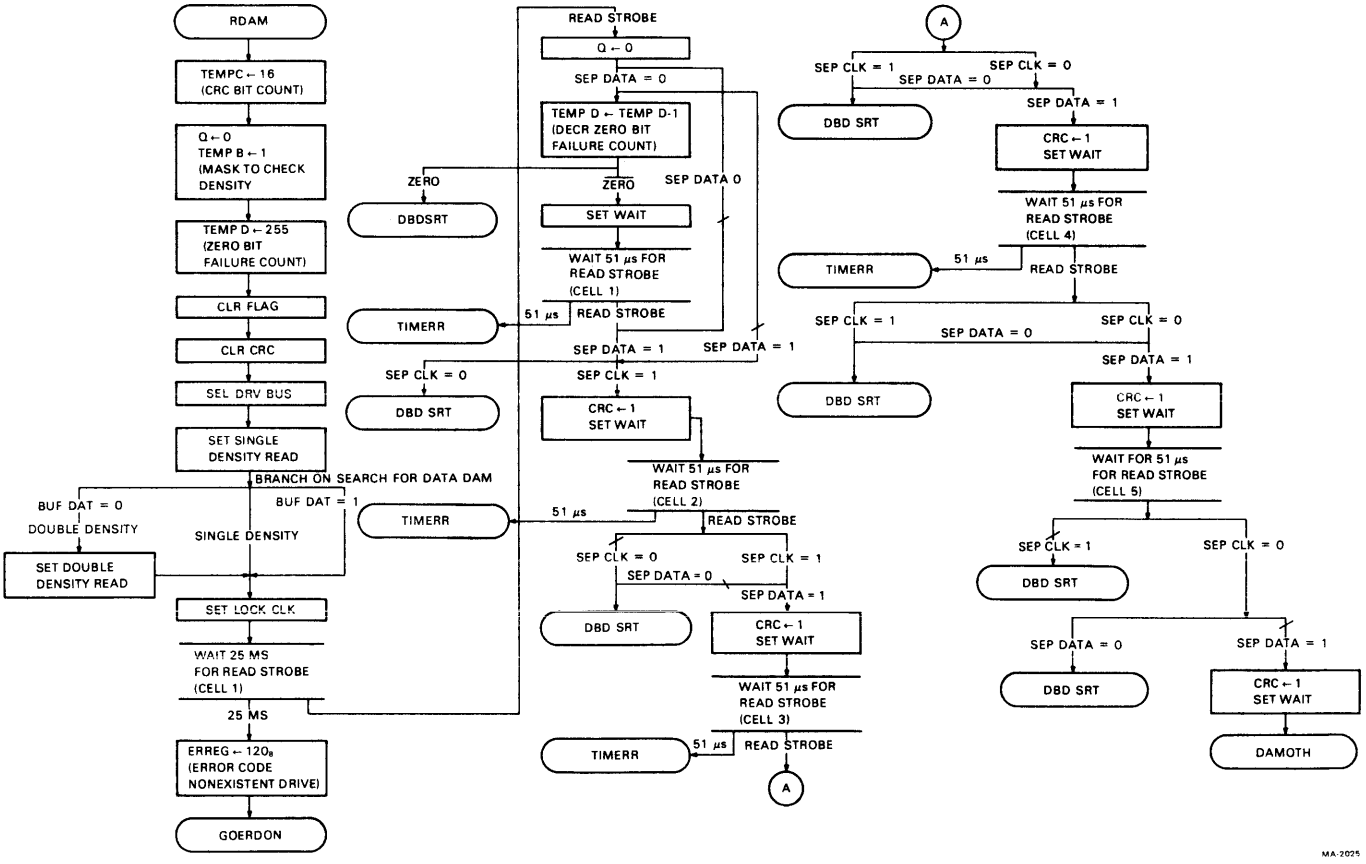


Figure 5-20 Read Address Mask Subroutine
Flowchart (Sheet 1 of 3)

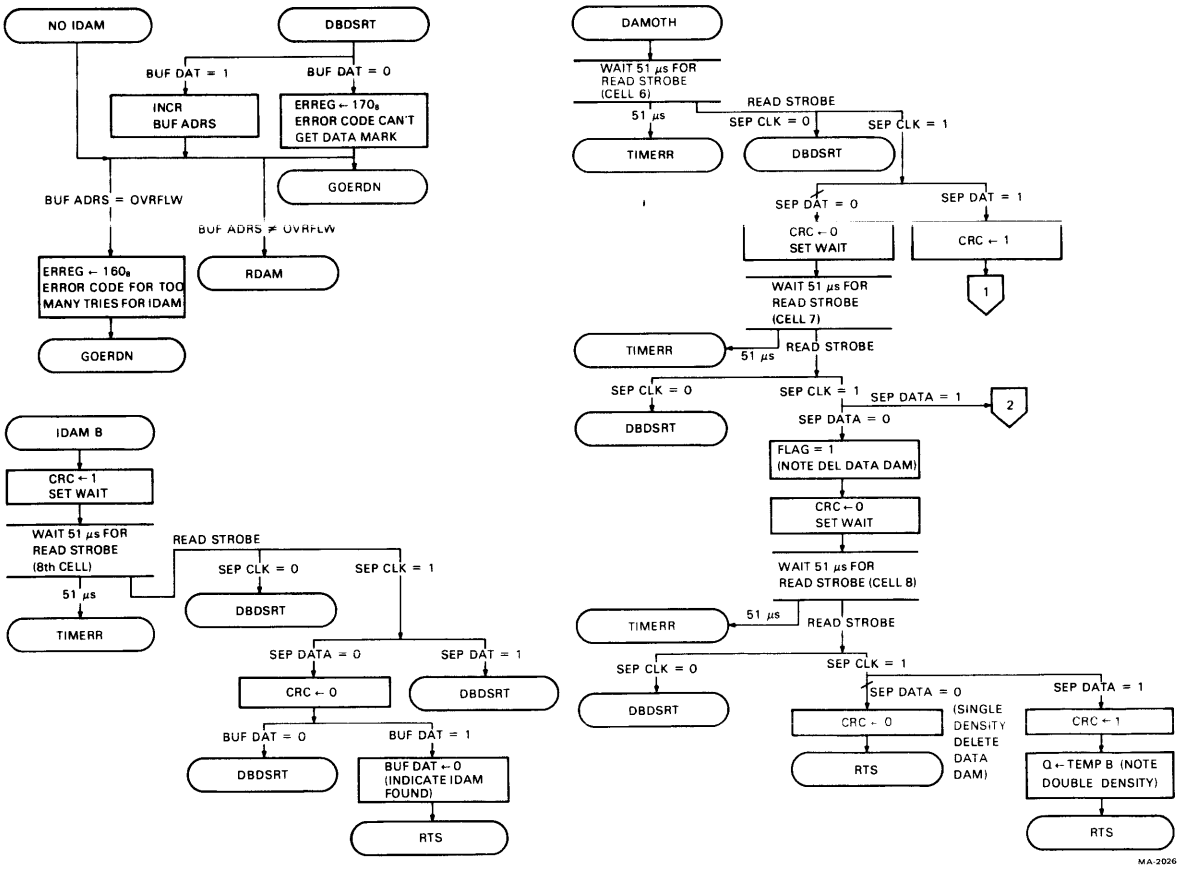


Figure 5-20 Read Address Mask Subroutine Flowchart (Sheet 2 of 3)

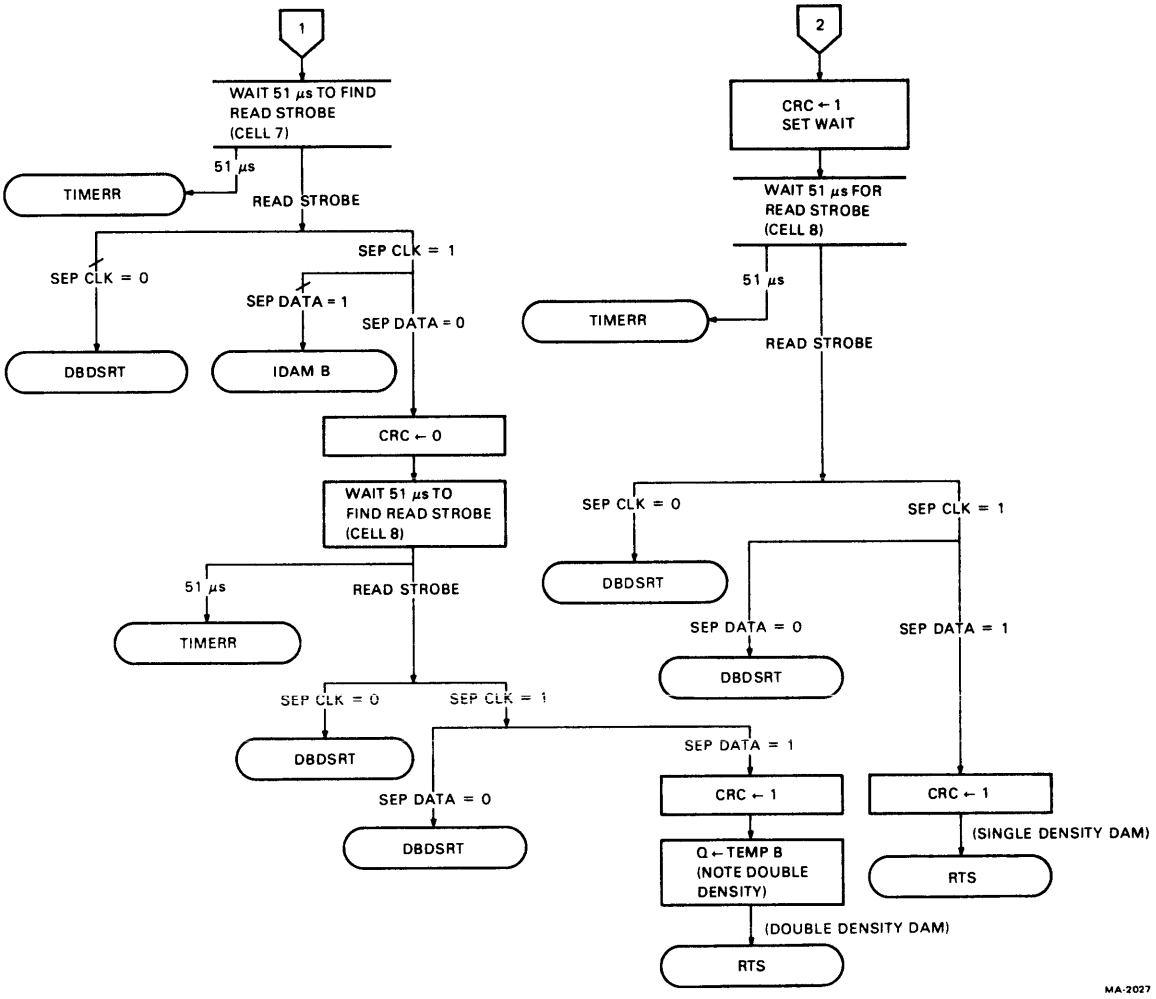


Figure 5-20 Read Address Mask Subroutine Flowchart (Sheet 3 of 3)

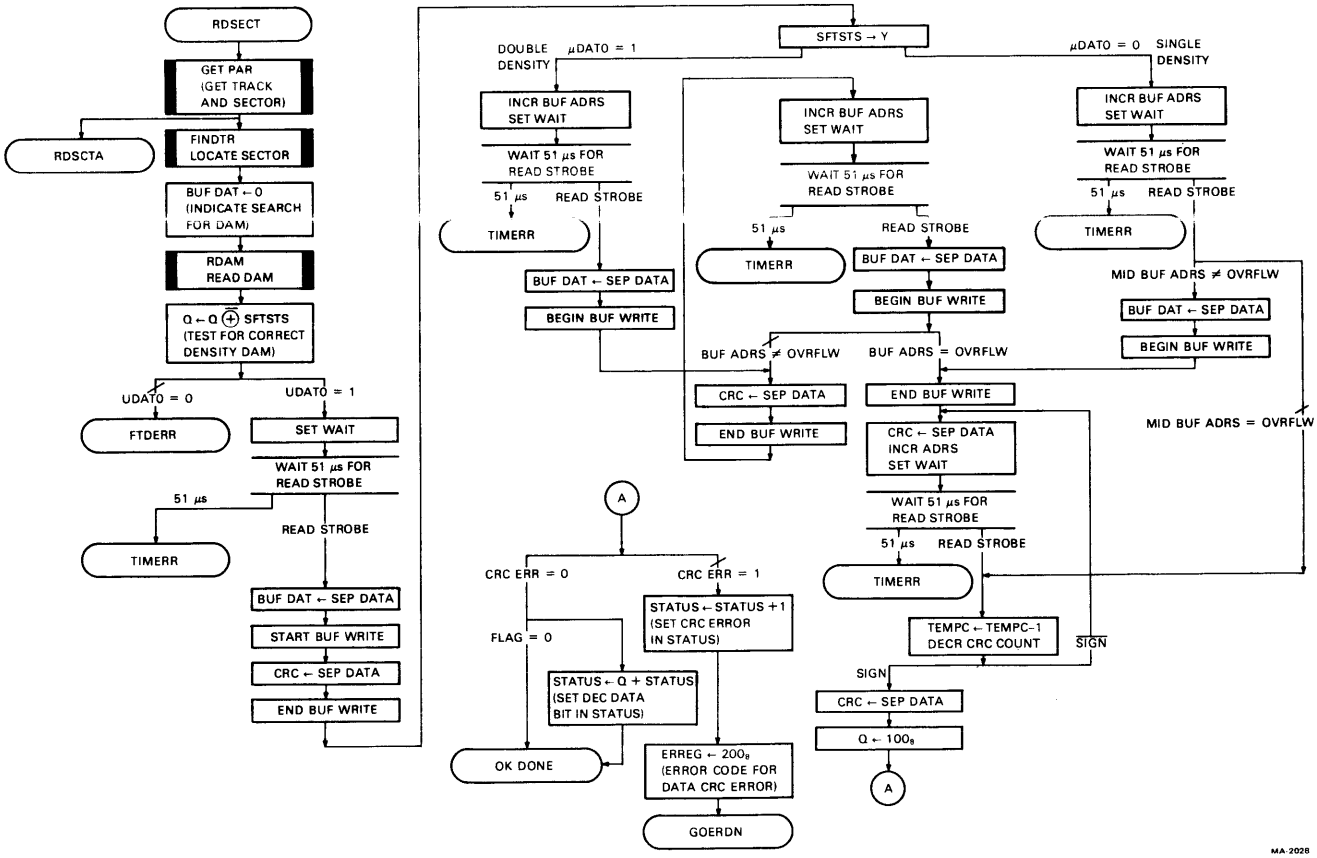


Figure 5-21 Read Sector Subroutine Flowchart

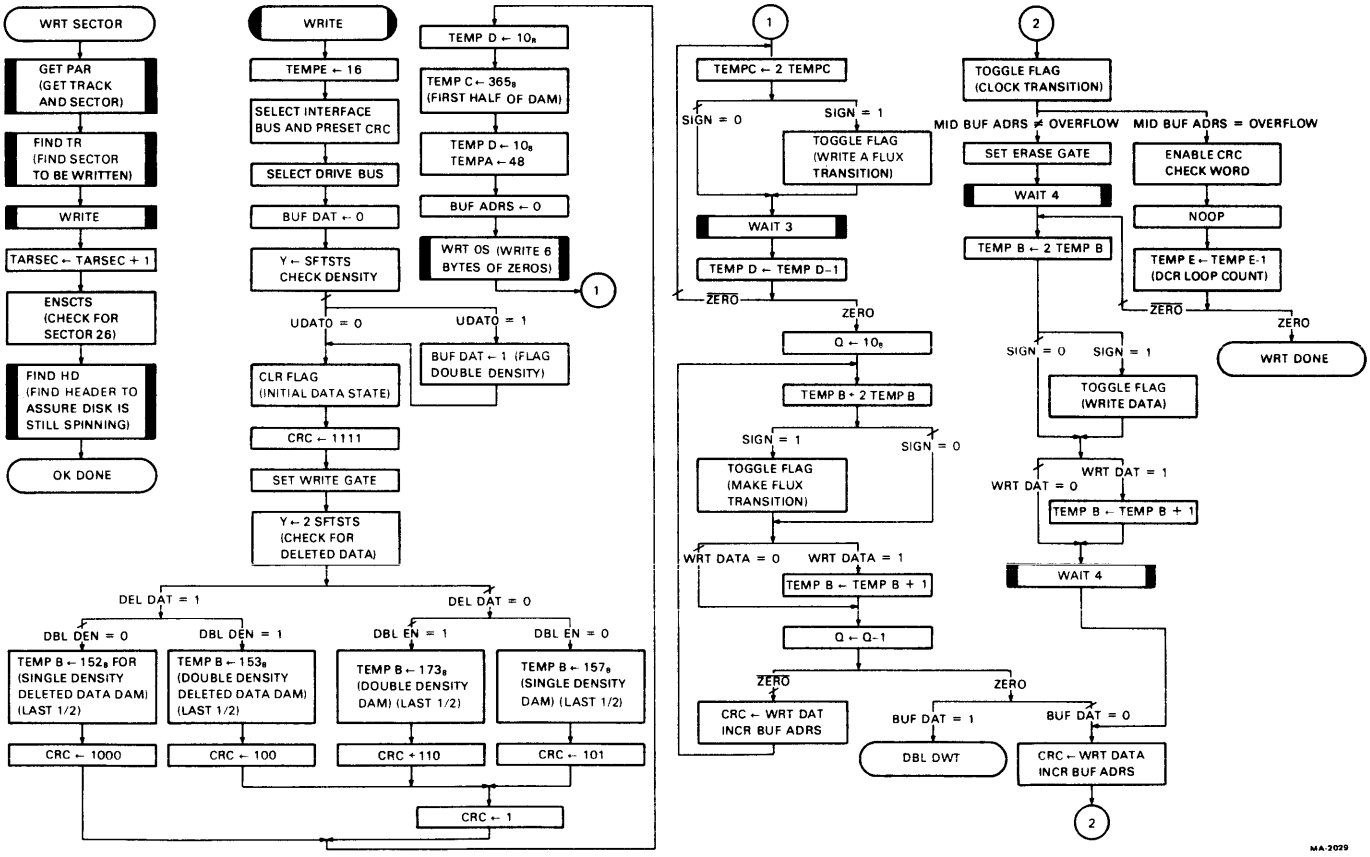


Figure 5-22 Write/Write Sector Subroutine Flowchart (Sheet 1 of 2)

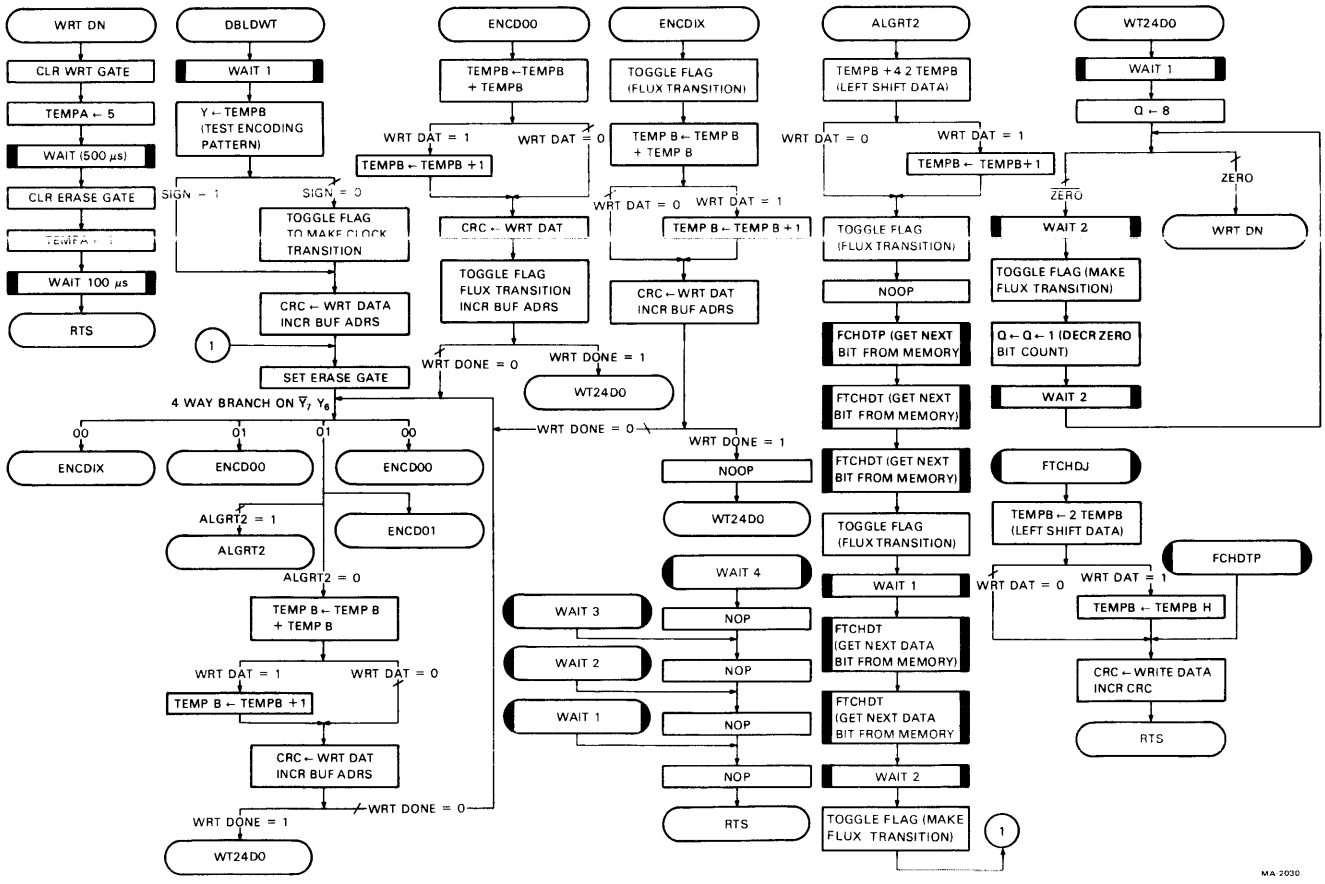


Figure 5-22 Write/Write Sector Subroutine Flowchart (Sheet 2 of 2)

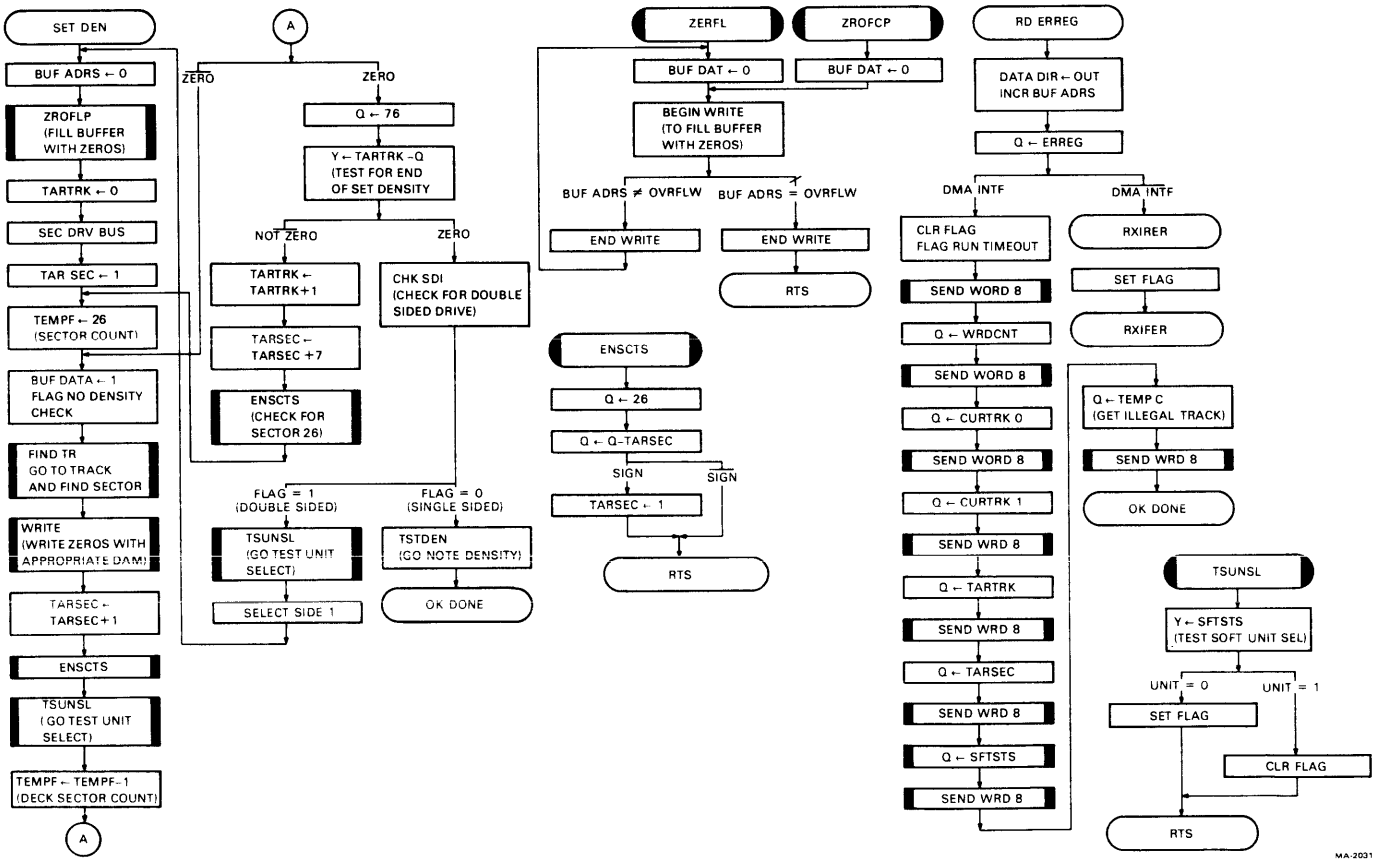


Figure 5-23 Read Error Register and Set Density Subroutines Flowchart

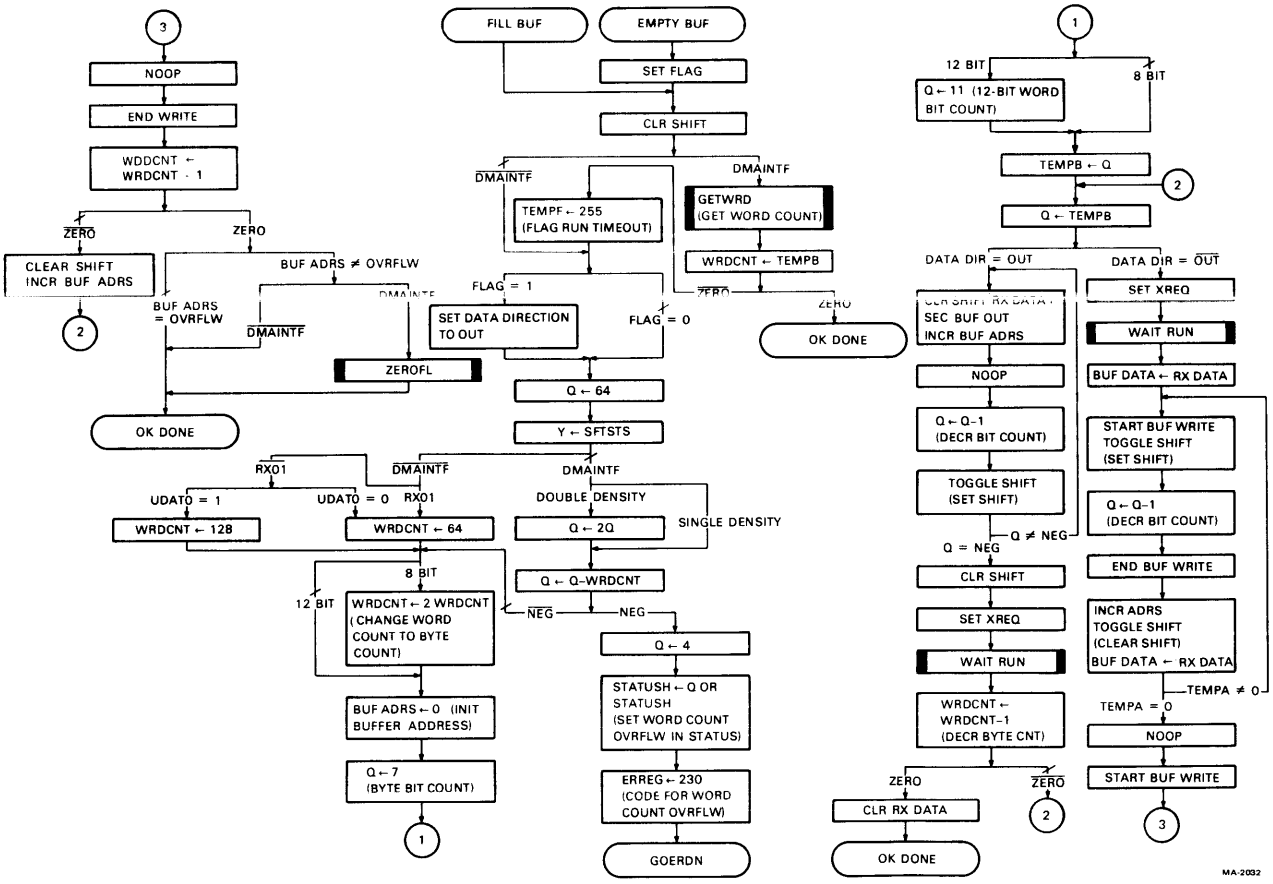


Figure 5-24 Fill/Empty Buffer Routine Flowchart

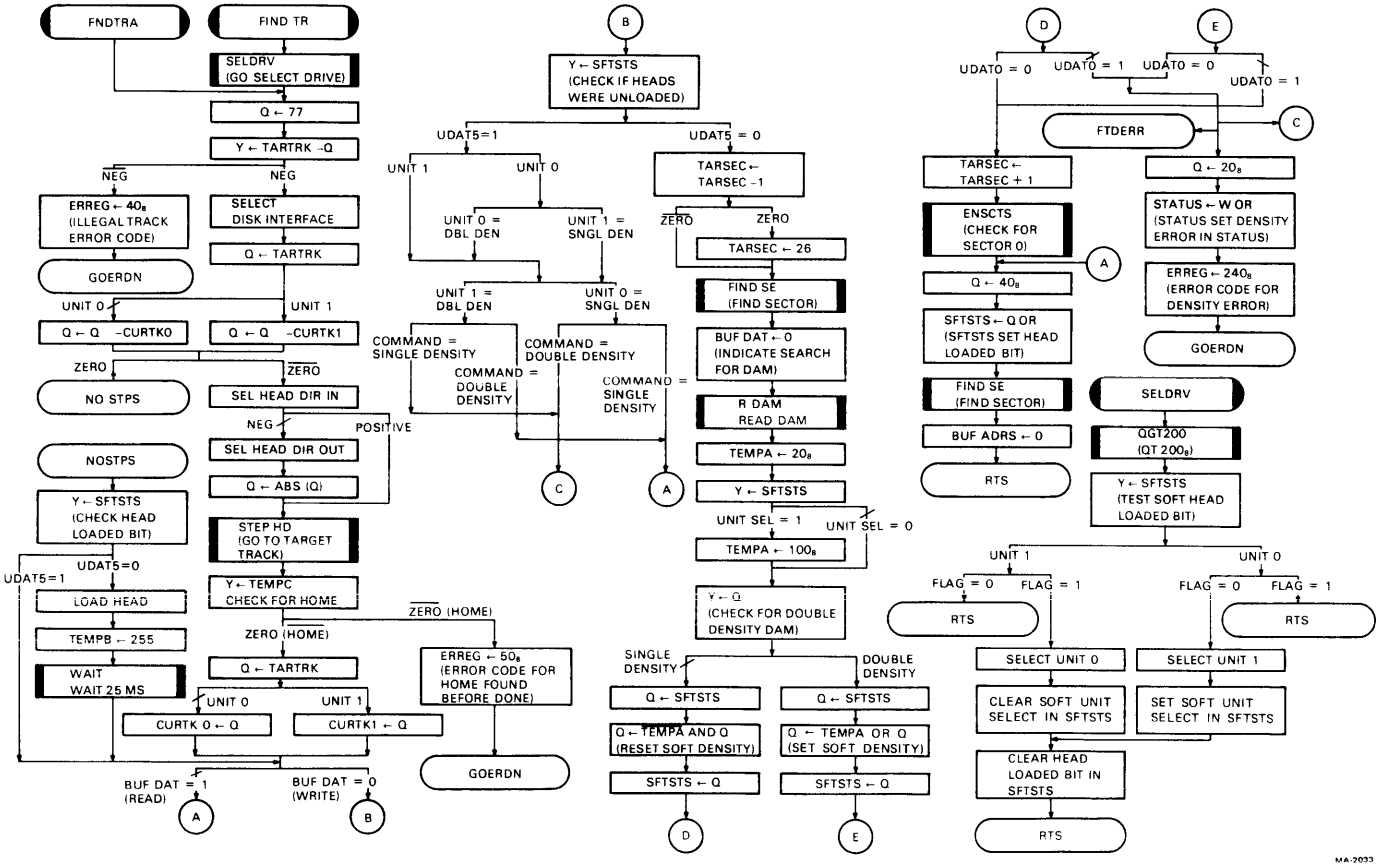


Figure 5-25 Select Drive and Find Track Subroutines Flowchart

The drive is determined and then the track address is checked to be sure it is a legal address. If the address is illegal, error (code 040) and done are set, but if the address is legal, the desired drive bus is selected and the direction for head movement is selected. The head is stepped to the target track. (If home is found before the target track, error – code 050 – and done are set.) On a write command if the heads have been unloaded, the DMA of the sector previous to the desired sector is read to determine if the diskette is of the appropriate density. If there is a density error, error and done are set (error code 240). After correct density is determined, the target sector is restored and the subroutine exits to the calling routine.

5.3.2.9 Decode command (DECCMD) Routine (Figure 5-26) – This routine which starts at microinstruction 1400 is used to decode the command from the host processor and subsequently to branch to the routine to perform the desired function. Transfer request is cleared, the direction of transfer is set to in, done is set, and the command is entered. The 8-bit or 12-bit mode is determined, done is cleared, and the function to be performed is decoded (write data, fill, etc.) and this routine jumps to the particular routine.

5.3.2.10 Maintenance Read Status (MRDST) and Check Ready (CHKRDY) Subroutines (Figure 5-27) – The MRDST and CHKRDY subroutines start at microinstructions 646 and 656, respectively. These subroutines are used to check the speed and density of the drive selected and to set the ready bit.

5.3.2.11 Get Parameter (GET PAR), Step Head (STEPHD), Wait, Wait Run, and Write Zeros (WRTS) Subroutines (Figure 5-28) – The GET PAR subroutine which starts at microinstruction 767 is used to store target track and sector. (The high order three bits of the 8-bit sector byte are masked.) The STEP HD subroutine starts at microinstruction 1741 and is used to step the read/write head to the target track; two step pulses are applied to the head motor to move the head one track. The WAIT subroutine which starts at microinstruction 1762 is used for delays in 100 μs multiples. The WAIT RN subroutine which starts at microinstruction 357 is used to time out (200 for RX02 DMA configurations, indefinitely for RX01 configuration and RX02 programmed I/O configurations) the wait for RUN to be asserted for data transfers to/from the interface module. If a timeout occurs, the routine asserts error and done. The WRTOS subroutine starts at microinstruction 326 and it is used to write zeros for the sync field (six bytes).

5.3.2.12 Find Sector (FINDSE), Send Word 12 (SNDW12), Send Word 8 (SNDW8), Get Command (GET CMD), and Get Word (GET WRD) Subroutines (Figure 5-29) – The FINDSE subroutine which starts at microinstruction 1543 is used to locate the target sector by comparing the current sector location of the read/write head with the target sector until a match is found, or if no match is made, an error (code 070) is set. The SNDW12 and SNDW8 subroutines start at microinstructions 1567 and 1621, respectively. These subroutines are used to send 8-bit or 12-bit words to the interface; for the 12-bit mode high order bits 8–11 are shifted so that they are the next four bits transmitted after the low order bits (0–7). The GET CMD and GET WRD subroutines start at microinstruction 752 and 1625, respectively. These subroutines are used to get the command (the length of the word in the command depends on 8-bit or 12-bit mode) and store it so the function contained in the command can be performed.

5.3.2.13 Maintenance Check Ready (MAINT CHK) Subroutine (Figure 5-30) – This subroutine which starts at microinstruction 721 is used to set maintenance mode and check the read/write electronics during maintenance (troubleshooting) by writing zeros and reading these same zeros to generate SEP DATA and SEP CLK signals. If an error occurs, error code 220 is set in the error register, and if no errors occur, maintenance mode is cleared and the routine returns to the calling routine.

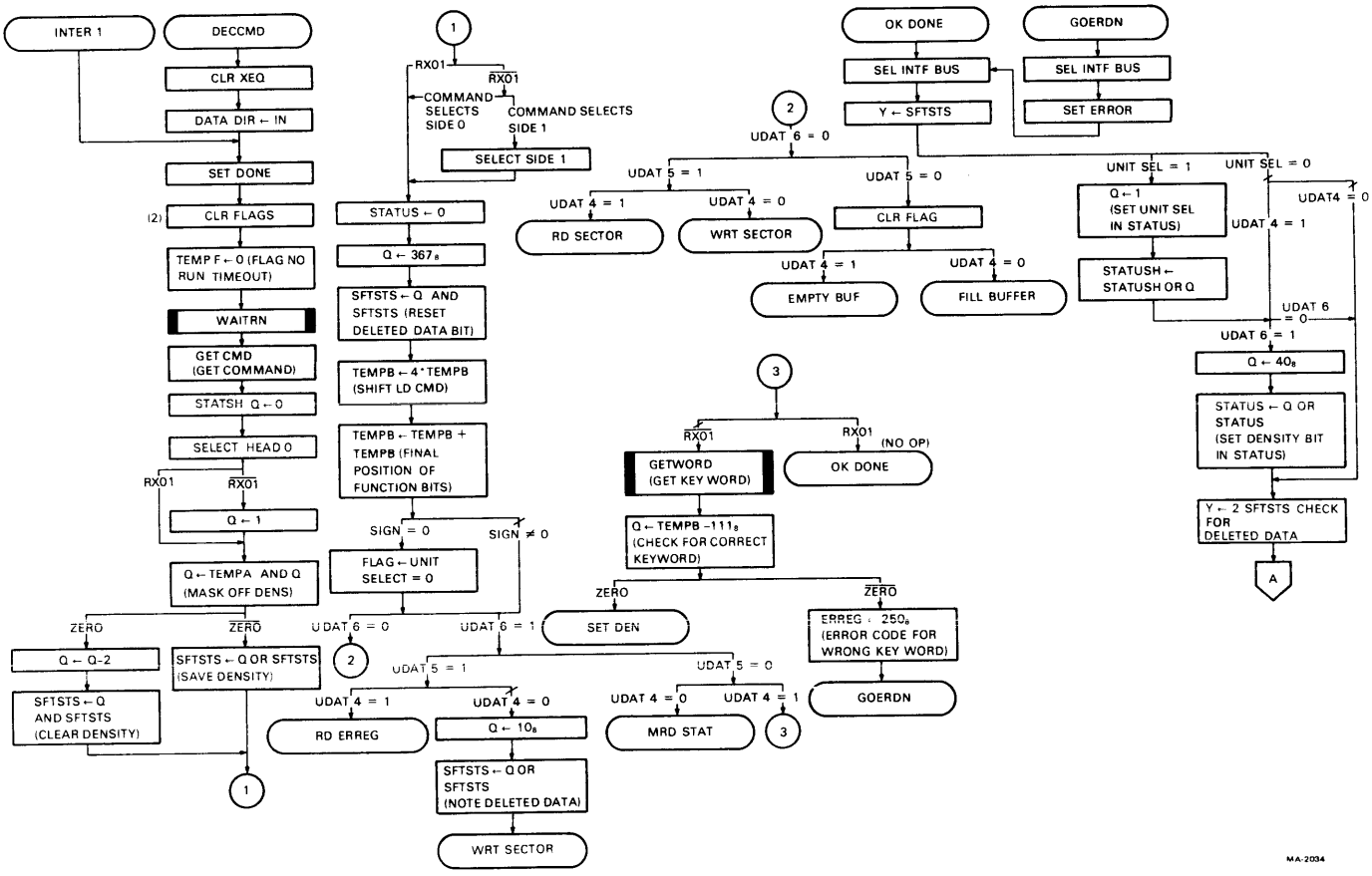


Figure 5-26 Decode Command Routine Flowchart
(Sheet 1 of 2)

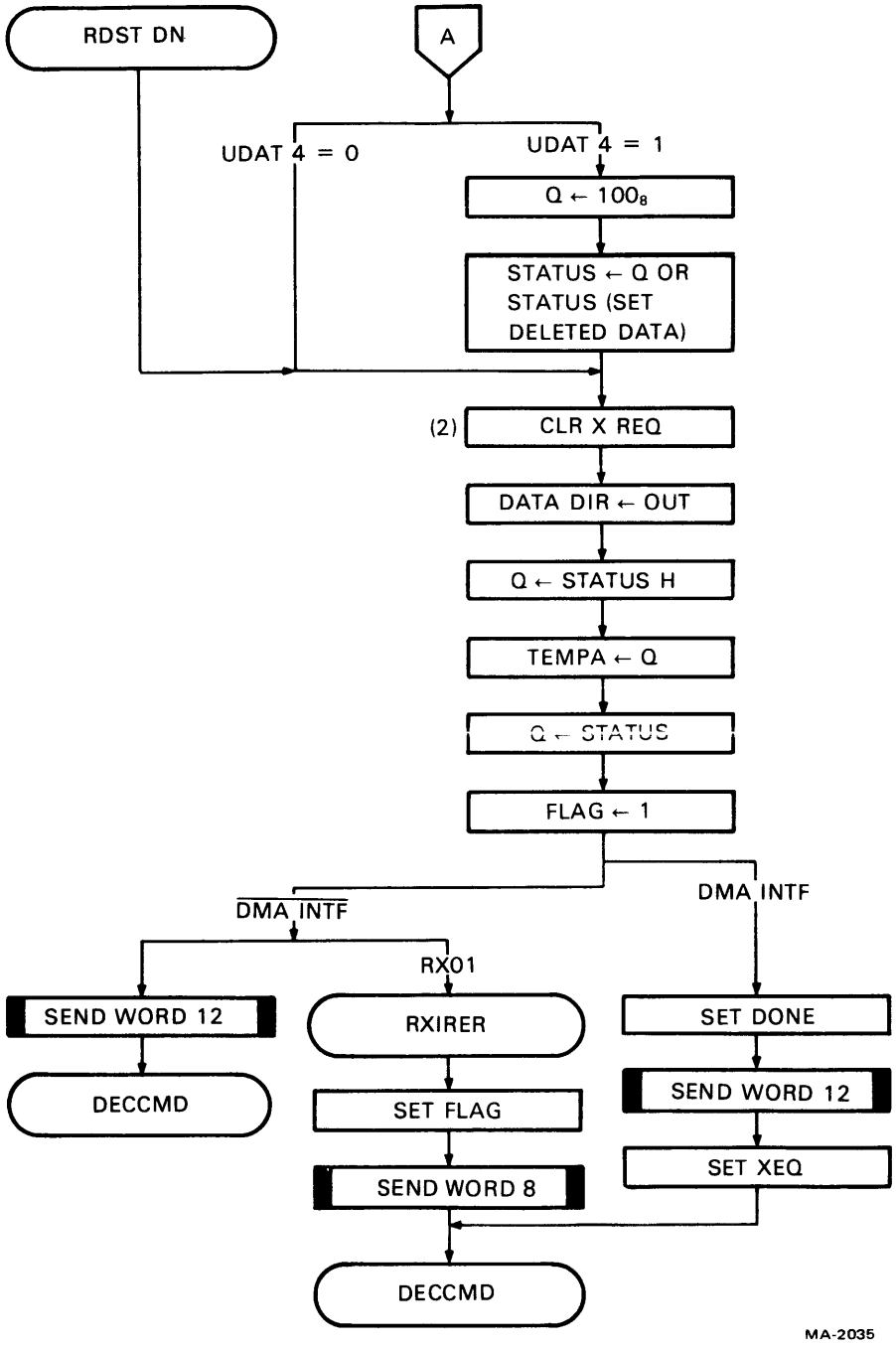


Figure 5-26 Decode Command Routine Flowchart (Sheet 2 of 2)

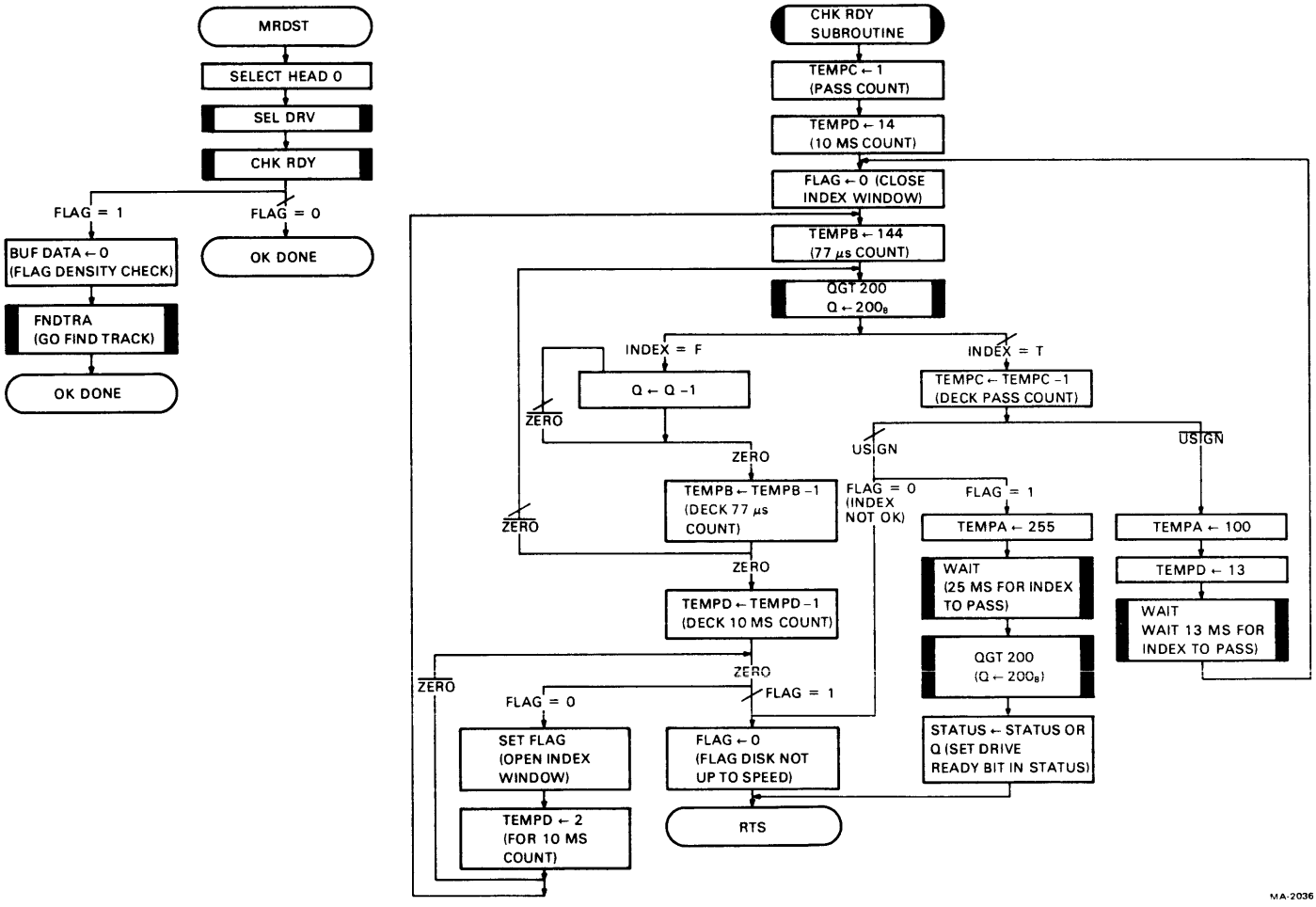


Figure 5-27 Maintenance Read Status and Check Ready Subroutines Flowchart

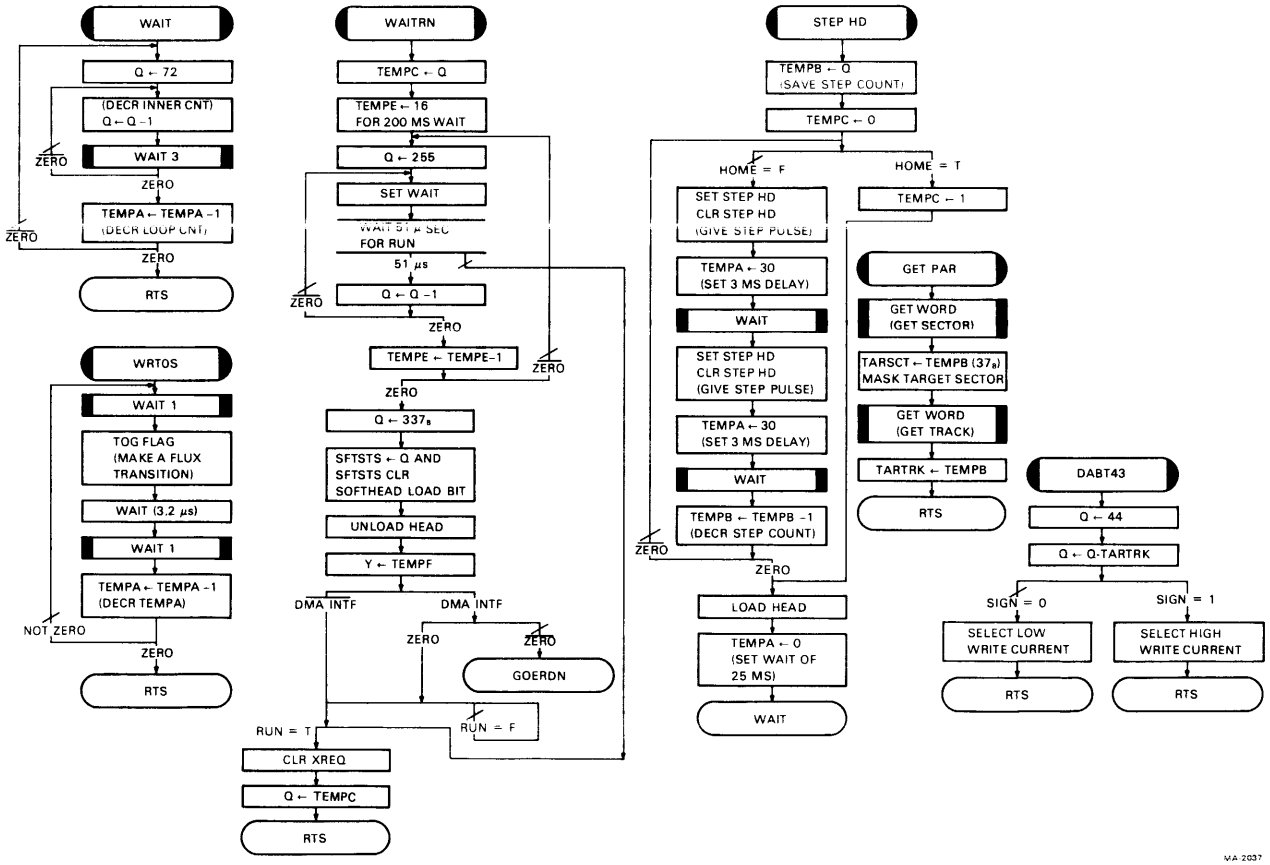


Figure 5-28 Get Parameter, Step Head, Wait, Wait Run, and Write Zeros Subroutines Flowchart

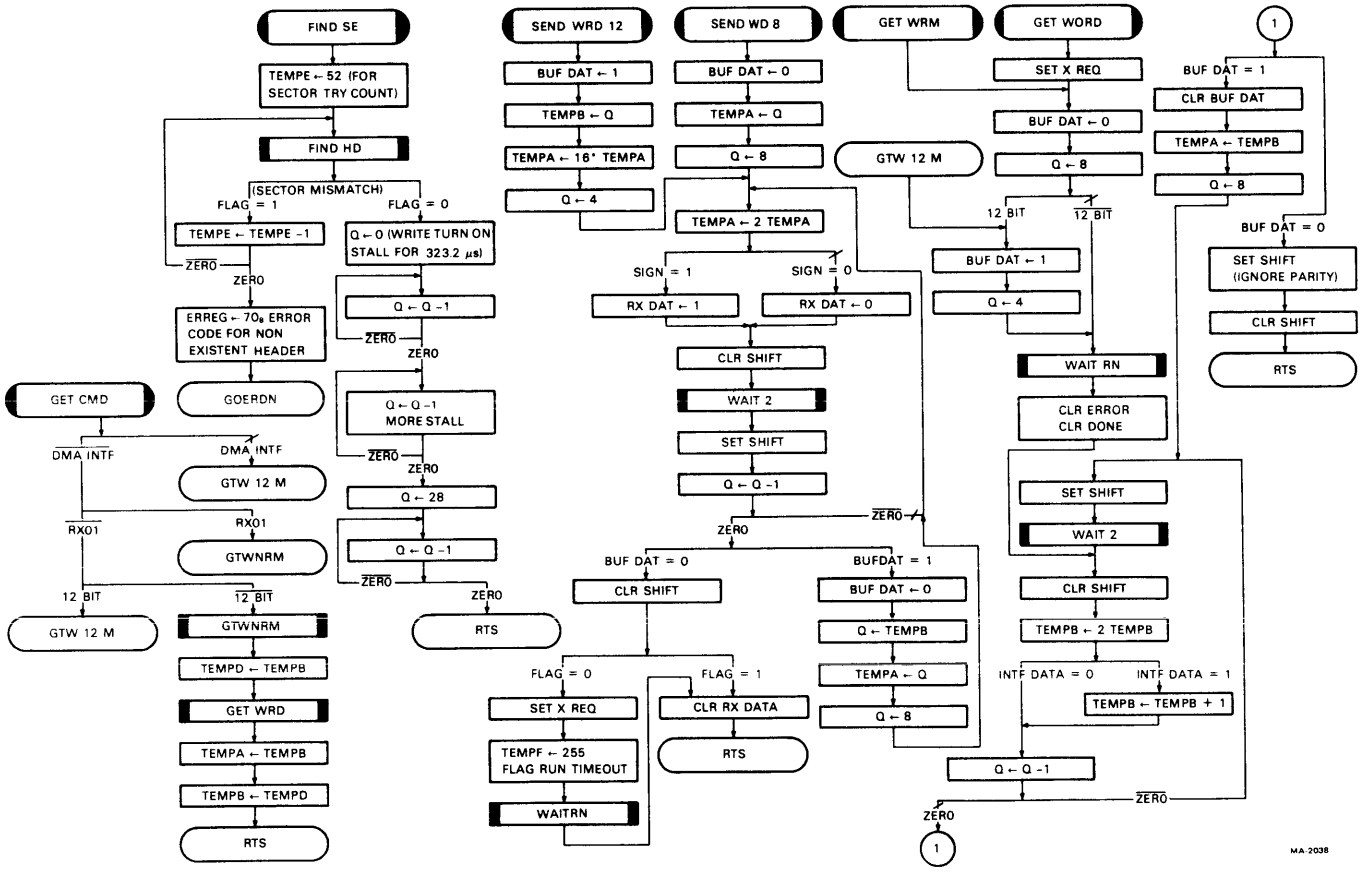


Figure 5-29 Find Sector, Send Word 12, Send Word 8, Get Command and Get Word Flowchart Subroutines

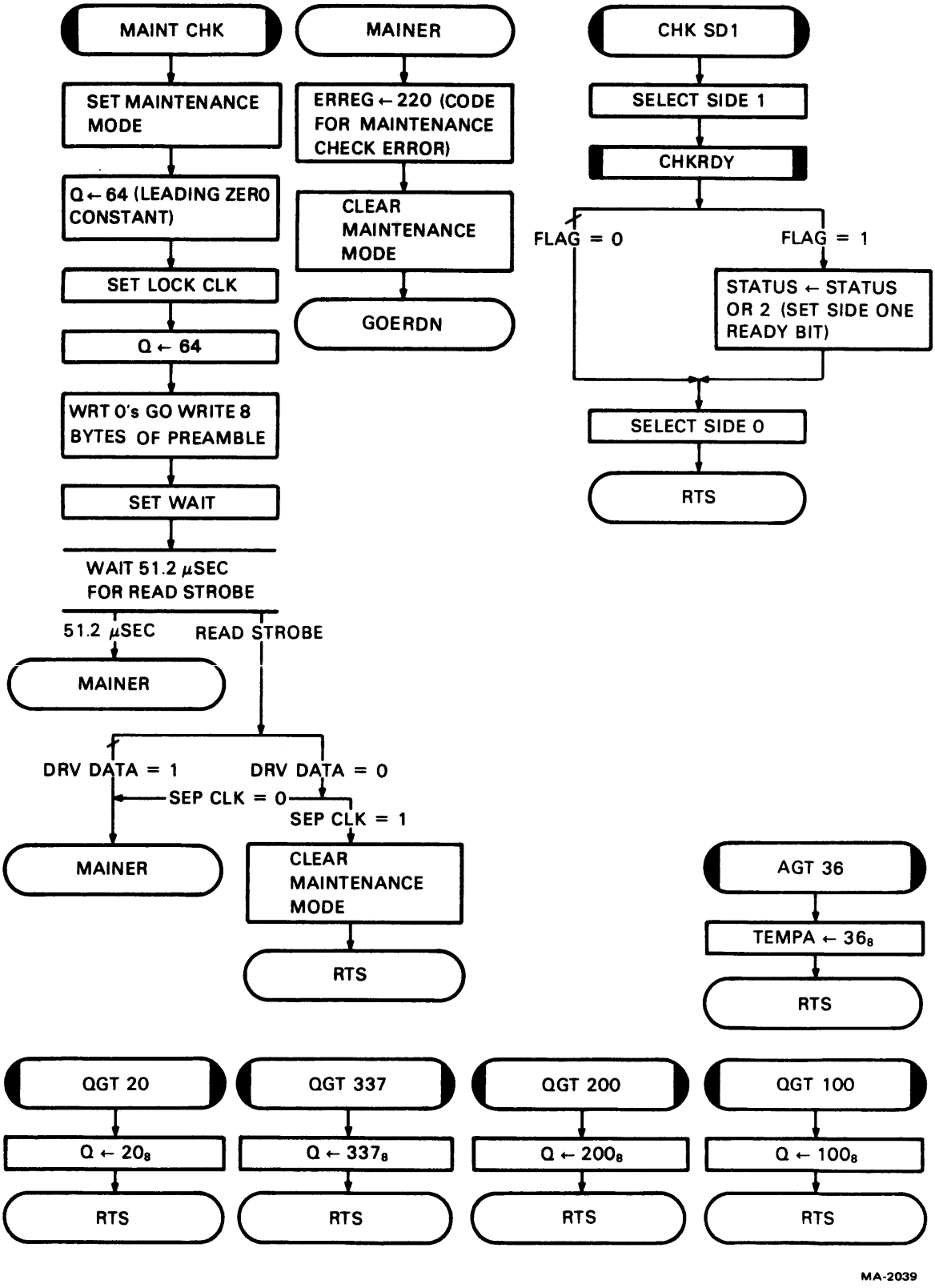


Figure 5-30 Maintenance Check Ready Subroutine Flowchart

5.3.3 Read/Write Block Diagram Description

The read/write electronics transfer data between the drive(s) heads and the controller. Data and control commands are processed so that the desired drive and head are selected and then data is either read from or written on the disk. Figure 5-31 is a block diagram of the read/write electronics which shows the signal flow to/from the major functional circuits of the assembly. The E references on the diagram are IC chip designations on the read/write print set.

5.3.3.1 Drive and Head Control – The drive and head control circuits consist of the head load control (E53, Q26, Q27) and the stepper motor control (E1 to E10, E57 to E61, Q12 to Q19). The head and drive selected are determined by DRV SEL DK1 which is applied to both circuits along with SEL DKO. The DRV AC L signal is used to initialize both the head load and motor control circuits; so when DRV SEL DKI is asserted, the drives and step counters for drive 1 motor are enabled and the step counting and driver circuits for drive 0 are disabled. The direction in which the head moves is determined by DRV OUT/ABOVE TRK 43; when it is asserted, the head moves out toward the edge and when the signal is negated, the head moves toward the center. With the drive selected, the head motor is stepped to the desired track using two step pulses (DRV STEP L) for each track moved. With the head positioned over the desired track, the DRV LOAD HEAD signal is asserted to activate the head solenoid for the drive selected by the assertion or negation of DRV SEL DKI. The drive and head selected status is applied to the controller by the position data selector.

5.3.3.2 Position Data Selection – The position data selector couples drive track 0 and head index data to the controller. Track 0 and head index hole detection are each accomplished by a LED phototransistor pair which develops a pulse each time the index hole passes or when the head is positioned over track 0. The track 0 signal for each drive and the index hole signal for the head(s) of each drive are applied to data selector E56. The outputs, DRV SEL TRK 0 and DRV SEL INDX, are determined by which drive (SEL DKO) and head (DRV SEL HDO) are being used. When SEL HDO and SEL DKO are both asserted, the DRV SEL TRK0 output is for drive 0, track 0 and the DRV SEL INDX output is for drive 0, head 0.

5.3.3.3 Read/Write Circuit – The read/write circuits consist of the head select matrix and read/write control, the write current amplifier, the read current amplifier, and the read data detector (PLL, data separator, preamble detector, data gate). These circuits process the data to be written on the disk or the data to be retrieved from the disk.

Data can be written on the disk when DRV WT GATE is asserted enabling the encoded DRV WT DATA to be coupled through the write current amplifiers to the drive head selected by SELDKO, SELDKI, SEL HDO. At the same time the write path is enabled by DRV WT GATE, the read path from the head select matrix is disabled. The DRV ERASE signal is asserted after DRV WT GATE to enable the erase amplifier in order to compensate for displacement between R/W pole and erase poles. The DRV OUT/ABOVE TRK 43 signal is used to disable a transistor in the write circuit and reduce the write current when writing on tracks above track 43; when closer to the center of the disk, bit density increases and less current is necessary.

Data can be read from the selected disk when DRV WT GATE is negated which enables the read path from the head select matrix (E27). The signal from the disk is amplified by the read current amplifier and filter (E25, E23) which enhances the signal and reduces high frequency noise; the analog playback signal is digitized by differentiating the signal and detecting the zero crossover. The digitized data is coupled through the data gate to the data separator so that the data can be retrieved. The data is separated by developing a clock signal which operates at the frequency of the playback and then using the clock signal to gate the data. The data gate couples the digitized data to the PLL, data separator, and preamble detector when MAINT MODE is negated. (When MAINT MODE is asserted DRV WWT DATA is coupled to the PLL, data separator and preamble detector in order to test the operation of these circuits.) For data acquisition, the PLL has to be aligned and locked onto the data frequency during the preamble (six sync bytes of zeros). The preamble detector which is enabled when DRV

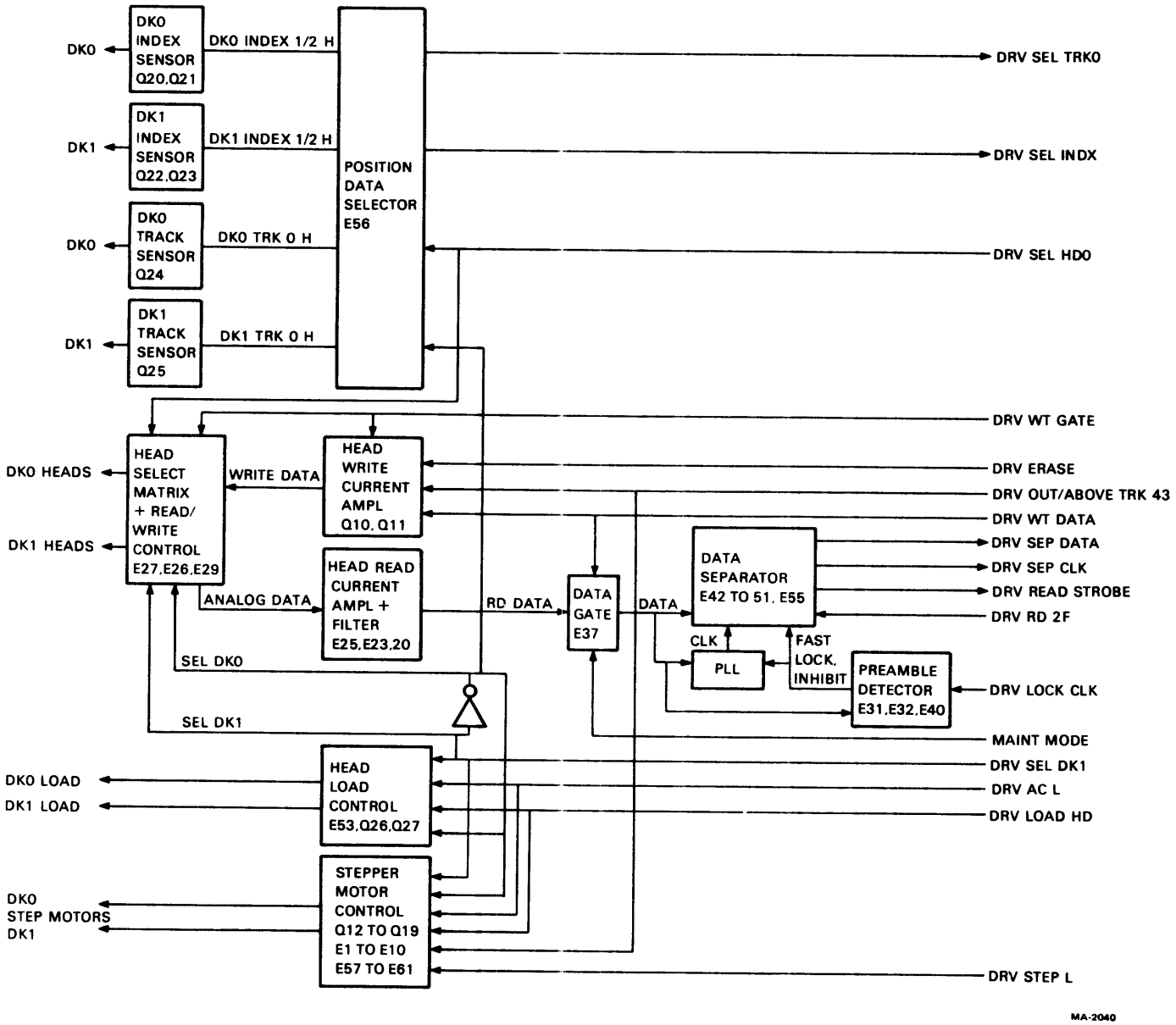


Figure 5-31 Read/Write Electronics Block Diagram

LOCK CLK is asserted by the controller counts the preamble bytes of zeros. After one byte of zeros, the PLL is enabled and begins to lock onto the data frequency. (The PLL generates a clock signal aligned to the data frequency by comparing the frequency of a VCO with the data frequency and adjusting the VCO until it approximates the data frequency.) The PLL locks onto the data frequency within four bytes of zeros unless a nonzero byte occurs which will reset the byte counter (E32, E40) and disable the PLL. After four bytes of zeros have been counted, FAST LOCK is negated which indicates that PLL is locked on. FAST LOCK enables the CLK and DATA to be processed by the data separator. The CLK signal is used to establish a window which will be high during data and low during a clock for both FM and MFM playbacks (Figure 5-32). A "1" bit in the playback when the window is high is set in the data latch and a "1" bit in the playback when the window is low is set in the clock latch. Data and clock bits are latched in E48, and synced in E50 and subsequently stored in the 3-bit clock register and the 3-bit data register E55. (The 3-bit register is necessary because of modified MFM encoding.) When data is MFM encoded, DRV RD 3F is asserted by the controller and the PLL clock and data decoding are adjusted accordingly. DRV READ STROBE is asserted as DRV SEP DATA and DRV SEP CLK are available at the output. The SEP DATA or SEP CLK signal is asserted for 4 μs for single density data and for 2 μs for double density data.

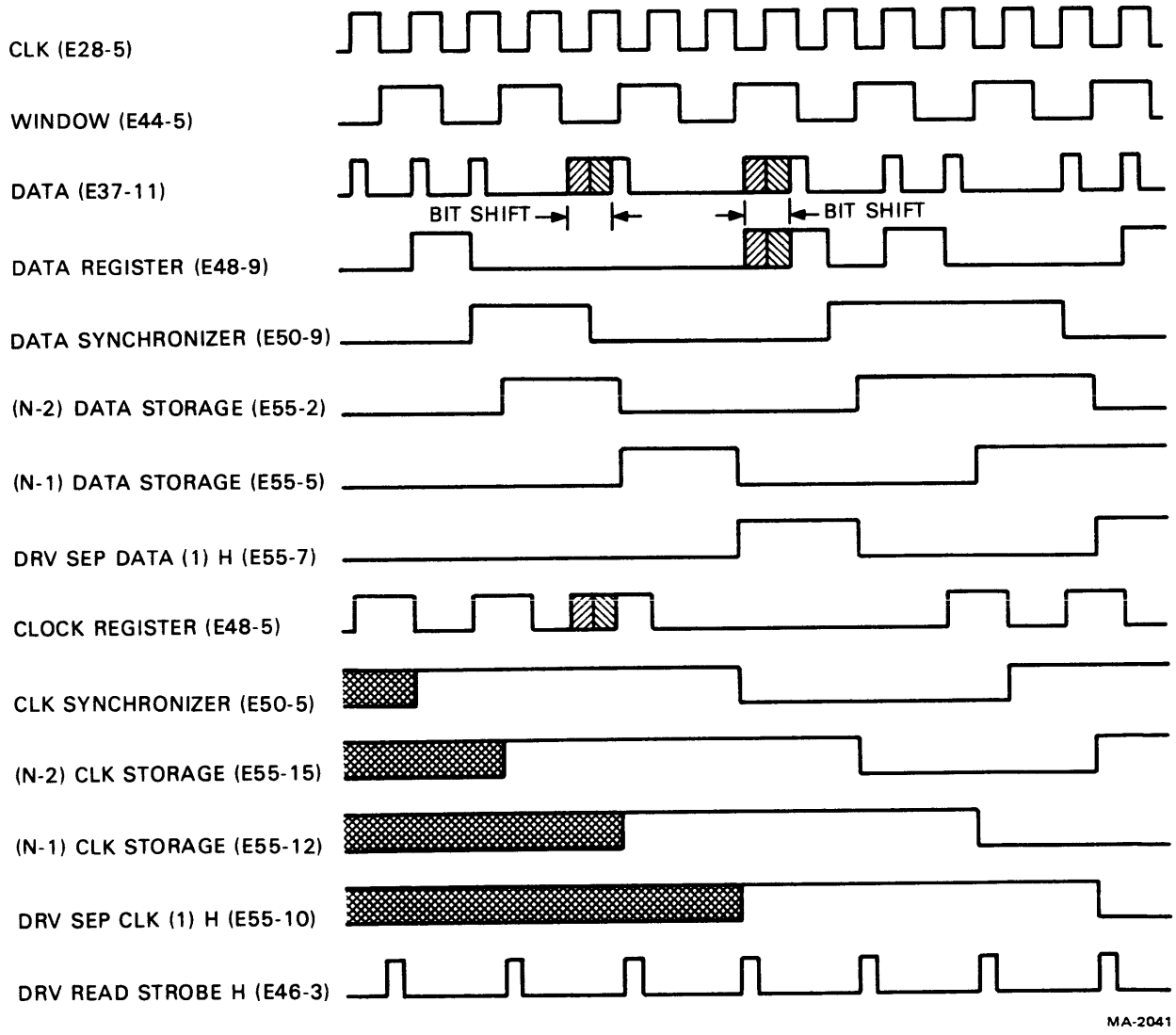


Figure 5-32 Data SYNC Timing Diagram

5.3.4 Mechanical Drive Description

The mechanical drive consists of four major parts:

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

The mechanical structure of the drive is shown in Figure 5-33; each section is described in the following paragraphs.

5.3.4.1 Drive Mechanism – The drive system provides rotational diskette movement using a single phase motor selected to match primary power of the controller system (Figure 5-34). The diskette drive attains ready status within 2 seconds of primary power application.

A cooling fan is mounted on one end of the drive motor shaft. Rotation of the diskette is provided by a belt and pulley connected to the other end of the motor shaft. The drive pulley and belt are selected for either 50 or 60 Hz power to achieve a diskette rotational speed of 360 rpm. (See Paragraph 2.1.3.2 for complete input power modification requirements.)

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

To load a diskette, the operator inserts the floppy diskette and presses down on the load handle which latches the load plate in the operating mode. The centering cone is mechanically linked to the load plate and is activated at the same time (Figure 5-35).

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

1. Engages the diskette media and drive mechanism.
2. Positions the diskette media in the correct track alignment.

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

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

5.3.4.3 Positioning Mechanism – The positioning mechanism comprises a carriage assembly and a bidirectional stepper motor (Figure 5-36). The stepper motor rotational movements are converted to linear motion by the rotor lead screw.

The read/write head mount rides on the lead screw and is held in horizontal alignment. When the stepper motor is pulsed, the lead screw rotates clockwise or counterclockwise, moving the mount in or out.

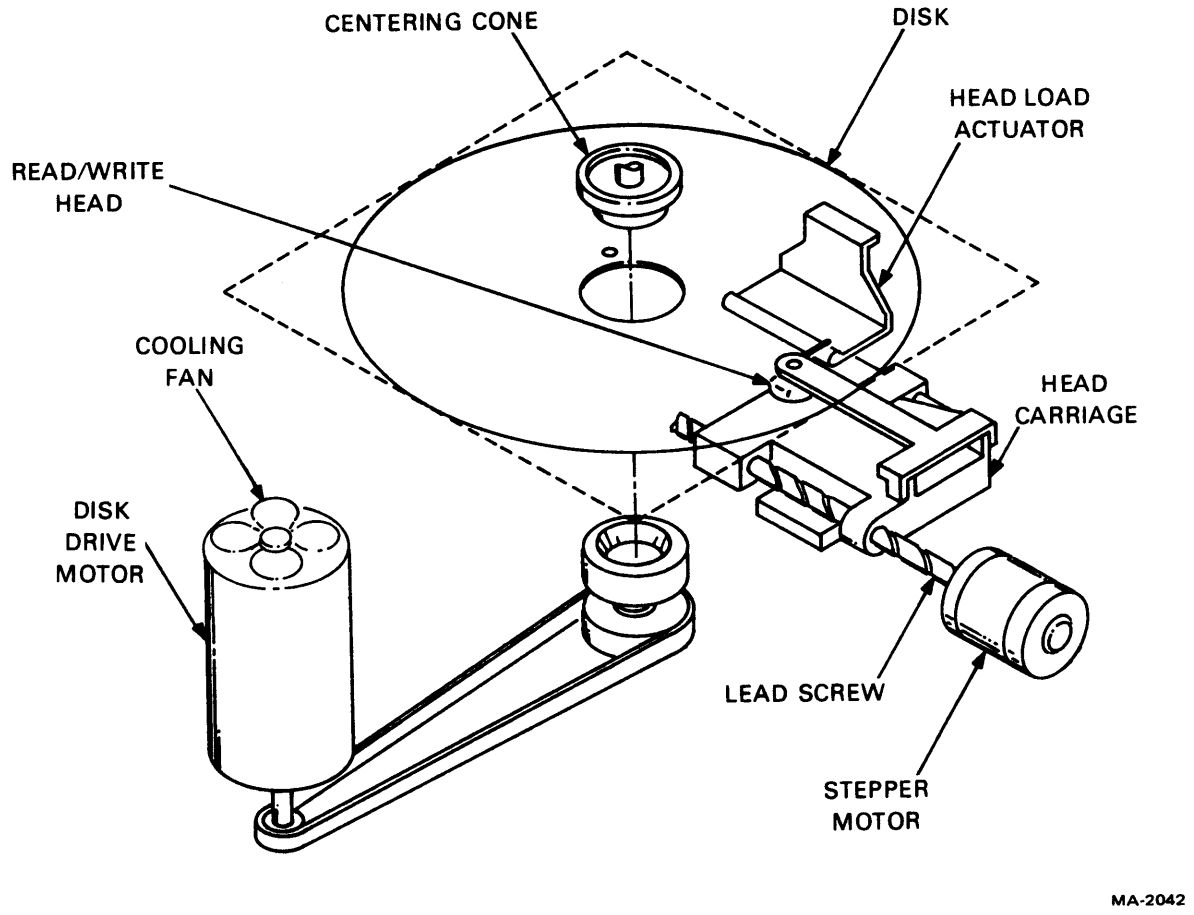


Figure 5-33 Disk Drive Mechanical System

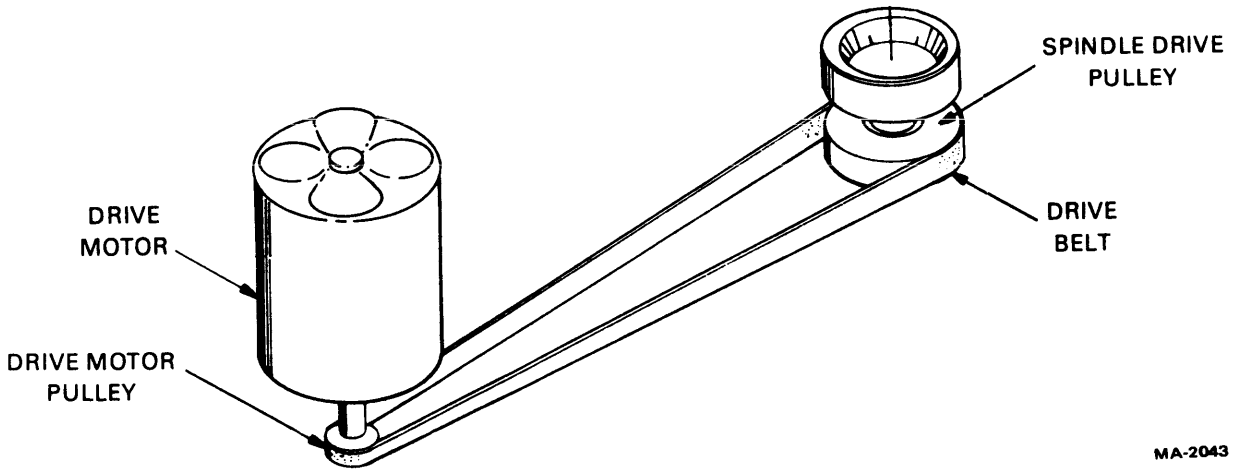


Figure 5-34 Drive Mechanism

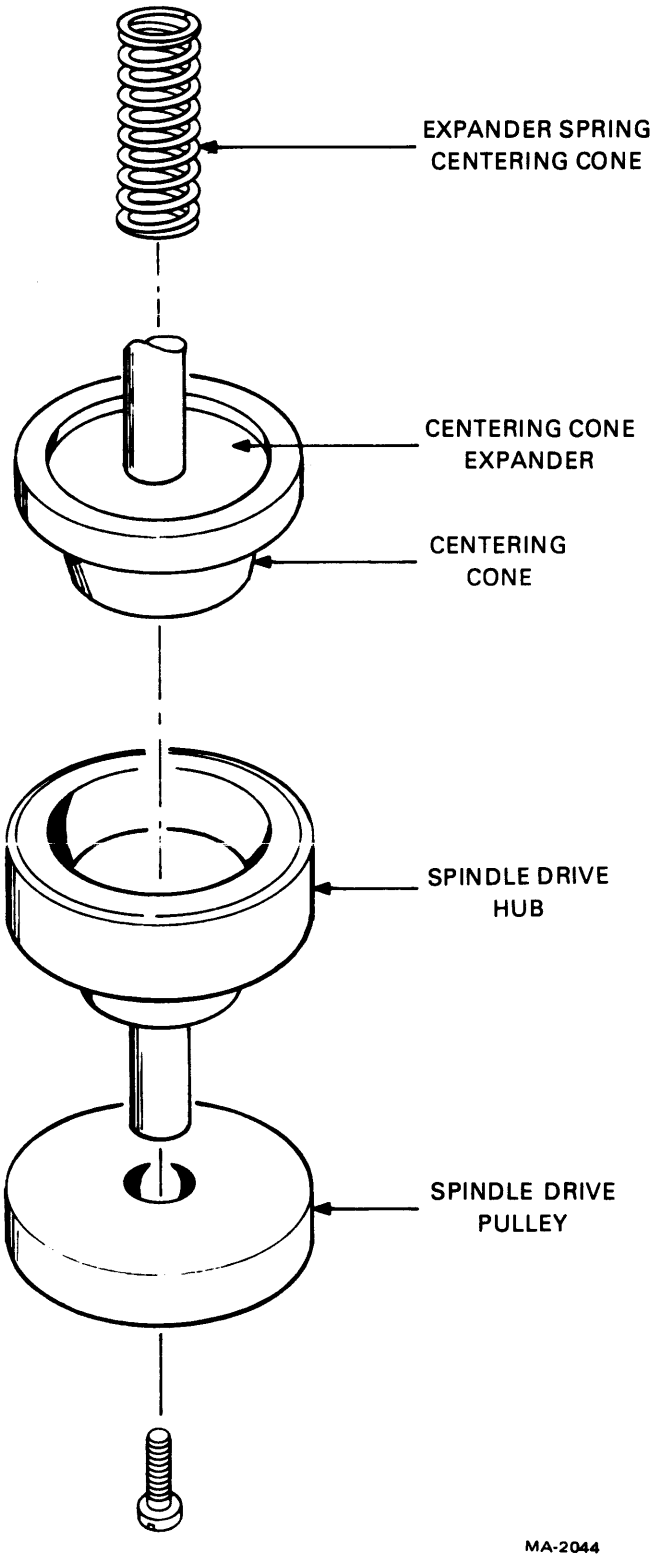


Figure 5-35 Centering Cone and Drive Hub

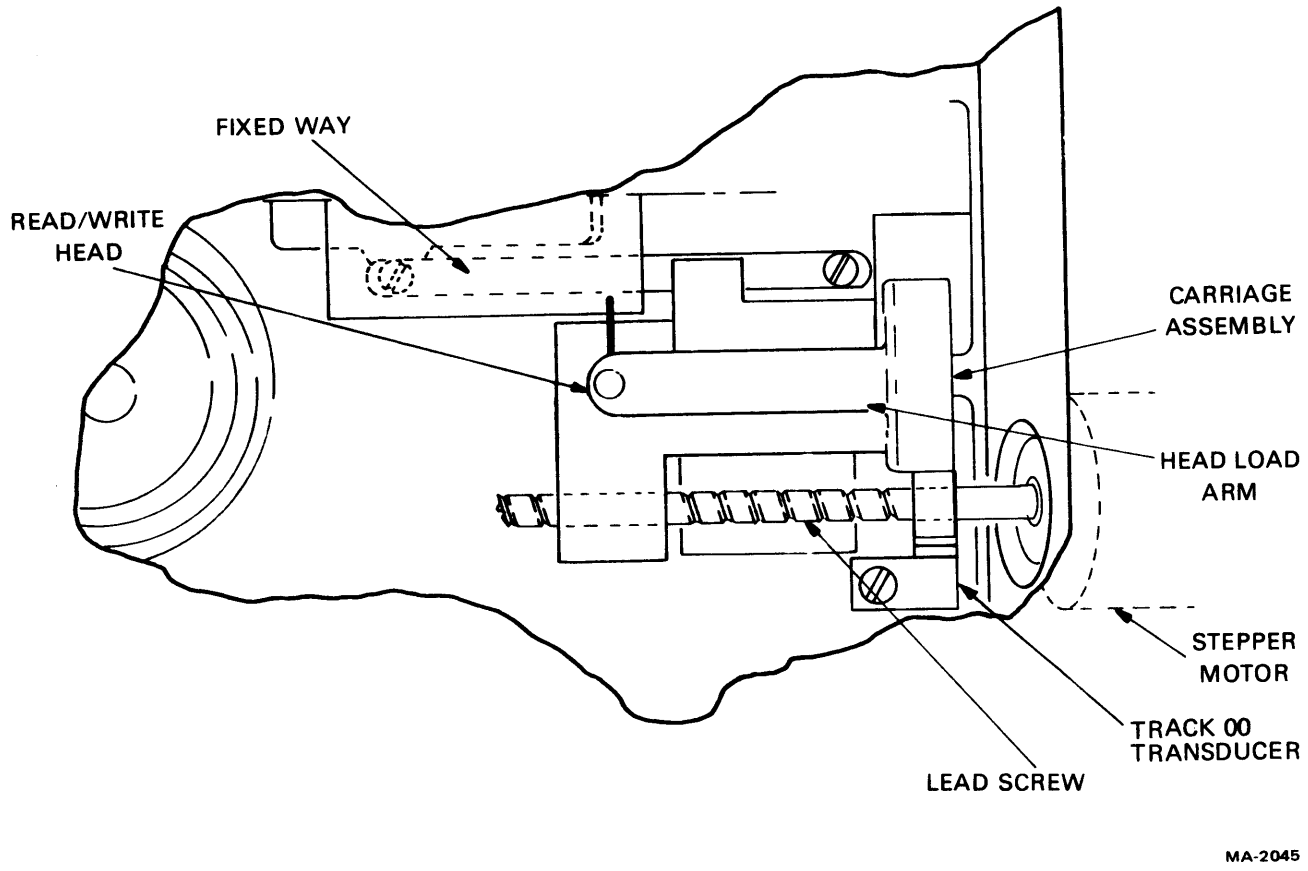


Figure 5-36 Positioning Mechanism

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

5.3.4.4 Head Load Mechanism – The head load mechanism is basically a relay driver and a solenoid. When activated by signal LD HD from the controller, the spring-loaded head load pad is released and rests in parallel alignment with the floppy diskette surface. The disk clamping platform (located on the casting) provides the lower alignment dimensional surface, while the head load solenoid bar provides the upper alignment surface.

In the load position, the read/write head rides between these two alignment surfaces and keeps the read/write head in contact with the diskette surface. The load pad is located behind the read/write head and holds the floppy diskette flat against the lower alignment block. To minimize diskette surface and head wear, the head is automatically disabled by the controller if no new command has been issued within 200 ms. (Head settling time is 25 ms.)

CHAPTER 6
MAINTENANCE

This chapter contains information necessary to maintain and troubleshoot the RX02 Floppy Disk System. Included is equipment care information, troubleshooting procedures, diagnostic routines, and removal and replacement procedures.

6.1 EQUIPMENT CARE

There is no scheduled preventive maintenance to be performed on the equipment. However, the exterior of the equipment should be kept clean using a damp cloth; if necessary the internal chassis should be cleaned with a vacuum cleaner.

NOTE

There are no field adjustments to be made on the RX02.

6.2 TROUBLESHOOTING THE RX02

Sections are included for troubleshooting the RX02 with and without the use of diagnostics. When attempting to diagnose the failing component the following procedure is recommended.

6.2.1 M7744, M7745 Failures

Place the replacement module on the existing M7745 module using a memory shield to prevent shorting. Remove the mounting hardware only when a module has been verified to be faulty.

6.2.2 Drive Failures

If one drive is believed to be faulty, interchange the cables of drive 0 and drive 1. If failures remain with a given drive number, replace module M7745 or M7744. Remove the drive from the chassis only if failures continue.

6.3 TROUBLESHOOTING WITH DIAGNOSTICS

Diagnostics consist of the RX02 Logic/Function Test, the RX02 Subsystem Performance Exerciser, and the DECX-11/DECX-8 modules. To load each diagnostic refer to the associated diagnostic documentation. Successful completion of the diagnostics will indicate that the system is operational. If error codes are displayed when running the performance exerciser or DECX refer to Table 6-2 to identify the probable cause of the error.

NOTE

The following procedure should only be used if the RX02 is the only loading device on the system and it is impossible to load the diagnostics.

6.4 TROUBLESHOOTING WITHOUT A DIAGNOSTIC

6.4.1 RX211 and RXV21 Systems

1. Install a known good diskette in drive 0
2. Initialize processor
3. Examine RX2CSR (777170): bits 11, 5 should be set; bit 15 should not be set.

   a. no bit 11 – replace interface
   b. no bit 5 – Table 6-1, Done bit timeout
   c. bit 15 set – Table 6-1, initialize error

Table 6-1 Troubleshooting Chart

Problem	Probable Cause	Suggested Remedy
No indication of power (drive inoperative)	Power cord not connected	Check power cord connection.
	Blown fuse	Check fuse; replace if blown.
	Drive power connector disconnected	Check power cable connection to drive.
	Power supply fault	Check power supply output: +24 Vdc at P1-6 + 5 Vdc at P1-4 – 5 Vdc at P1-2 } R/W electronics + 5 Vdc at P2-4 +10 Vdc at P2-1 } Controller Replace power supply if voltages not correct.
Drive not ready	Door open	Close door.
	Diskette not spinning	Replace belt.
	Motor not rotating	Check ac power connector.
	M7745 module	Replace module.
	M7744 module	Replace module.
Cannot run diagnostics	Improperly loaded	Reload diagnostic.
	Interface problem	Replace interface.
	Possible CPU problem	Run CPU diagnostics.

Table 6-1 Troubleshooting Chart (Cont)

Problem	Probable Cause	Suggested Remedy
Initialize Error	Interface cable inserted backward	Reverse cable.
	Diskette problem	Replace diskette.
	Switches not set properly	Check switch settings on M7745.
	M7744 module	Replace M7744.
	M7745 module	Replace M7745.
Done bit timeout	Switches not set properly on M7744 module	Check switch settings on M7744.
	M7744 module	Replace M7744.
	Interface module	Replace interface.
TR bit timeout	M7744 module	Replace M7744.
	Interface module	Replace interface.
Set Density Error	Diskette M7745 M7744	
Data error but no CRC error	M7744	Replace M7744.
	Interface	Replace interface.
For RX211 and RXV21 only	Interface	Replace interface.
Fill Buffer Error	NPR jumper on PDP-11 backplane not cut	Cut jumper.
Empty Buffer Error	Interface	Replace interface.
	M7744 module	Replace M7744.
Unexpected interrupt to loc. xxx	Switch settings on interface	Check switch settings on interface.
	Interface	Replace interface.

4. Examine RX2 ES (777172): bits 2, 7 should be set; bits 0, 1, 3, 4, 11 should not be set.

   a. no bit 2 – replace M7744
   b. no bit 7 – Table 6-1, Drive not ready
   c. bit 0 set – Table 6-2, Code 200
   d. bit 1 set – replace M7744
   e. bit 3 set – Table 6-1; no RX02 power
   f. bit 4 set – replace diskette
   g. bit 11 set – replace interface.

5. If neither address answers, replace the interface module or check switches (jumpers) on the interface.

NOTE

If RX2CS bit 15 is set and all bits of RX2EX are valid, load and execute the following program to obtain the definitive error code.

```
001000  012701          MOV     #ADR,R1
        177170
001004  032711  1$:     BIT     #40,(R1)
        000040
001010  001775          BEQ     1$
001012  012711          MOV     #17,(R1)
        000017
001016  032711  2$:     BIT     #200,(R1)
        000200
001022  001775          BEQ     2$
001024  012761          MOV     #2000,2(R1)
        002000
        000002
001032  032711  3$:     BIT     #40,(R1)
        000040
001036  001775          BEQ     3$
001040  000000          HALT
```

Refer to Table 6-2 to correlate the definitive error code with the probable cause of error.

Table 6-2 Error Code Probable Causes

	Error Code															
000	**010**	**020**	**040**	**050**	**070**	**110**	**120**	**130**	**150**	**160**	**170**	**200**	**220**	**230**	**240**	**250**
If Error Bit is set.	D B C	D B C	E B	C D B	A C D B E	A C D B	A C D B G E	A C B	C D B	C D B	A C B	A C B	C B	E B	A B E C	E B

A = Diskette
B = M7744 controller
C = M7745 R/W electronics
D = Drive
E = Interface module
F = Cables
G = Power supply

6.4.2 PDP-8 and CM05-8* Based Systems

1. Insert a known good diskette into drive 0.

2. Load and execute the following program to obtain the contents of the RX2ES and the definitive error codes.

```
200/    6007    CAF
201/    6755    SDN
202/    5201    JMP-1
203/    6752    XDR
204/    3214    DCA 214
205/    1216    TAD 216
206/    6751    LCD
207/    6755    SDN
210/    5207    JMP-1
211/    6752    XDR
212/    3215    DCA 215
213/    7402    HALT
214/       0    ERROR AND STATUS REGISTER
215/       0    DEFINITIVE ERROR CODE, BITS 4
216/      16    (4:11)
```

LOC 214 contains the RX2ES
LOC 215 contains the definitive error code bits (4:11) (Table 6-2).

RX2ES bits 4, 8, 9 should be set; bits 7, 10, 11 should not be set.

a. no bit 9 – replace M7744; replace PDP-8 interface.
b. no bit 8 – verify switch settings on M7744; replace M7744; replace interface module.
c. no bit 4 – Table 6-1 (Drive not Ready)
d. bit 11 set – Table 6-2 (Code 200 Error)
e. bit 10 set – replace M7744; replace M7745
f. bit 7 set – replace diskette.

If the program does not run, the M7744 controller module should be replaced.

6.5 REMOVAL AND REPLACEMENT

The following steps define the procedures for removing and replacing the subassemblies of the RX02 Floppy Disk System.

6.5.1 Module Replacement Procedures

Floppy Disk Controller M7744 (Figure 6-1)

1. Remove power from the RX02.

2. Unscrew the three screws on the module and raise the module to the servicing position.

3. Remove the plugs in connectors J1 and J2.

* For VT78 and VT178 consoles the MR78 Field Service ROM package is required to boot ODT into the console prior to entering the short diagnostic program.

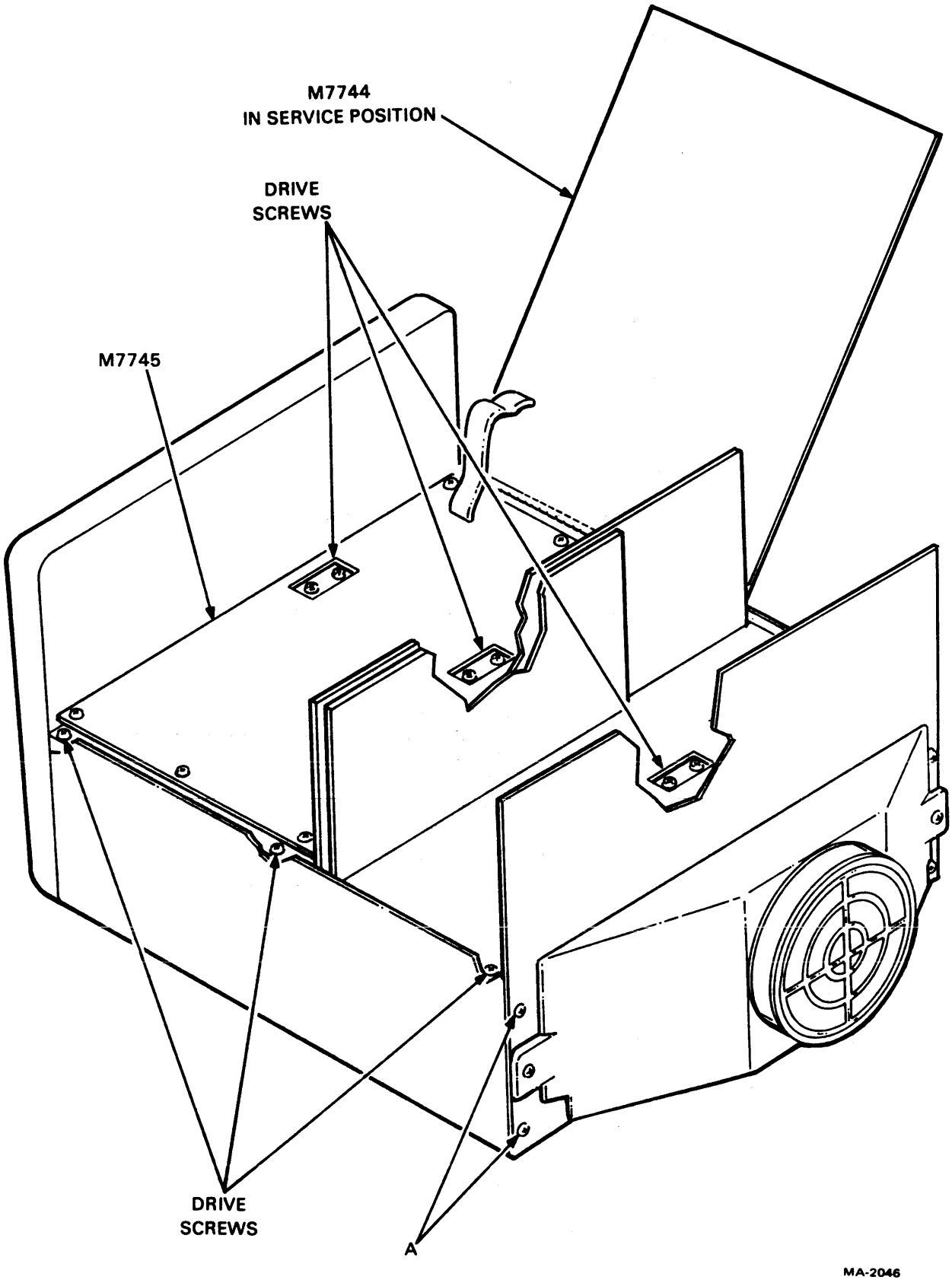


Figure 6-1 RX02 Component Location Diagram

4. Lower the module and remove the three screws holding the module onto the hinge.

5. Remove the module.

6. To install a module, perform steps 1–5 in reverse.

7. Ensure that cable BC05L-15 is inserted with the red stripe on the cable nearest the front of the unit.

Read/Write Control, M7745 (Figure 6-1)

1. Remove power from the RX02.

2. Raise the floppy disk control module to the servicing position.

3. Remove the plugs from connectors on the module, ensuring that they do not drop into the drive.

4. Remove the eight screws holding the module to the frame and remove the module.

5. To install a M7745, replace the screws and plugs removed in steps 3 and 4, ensuring that they are reinstalled into the correct connector (Table 6-3).

Table 6-3 M7745 Connectors

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

H771 Power Supply Regulator, 70-10718

1. With the power off, remove the plug from the regulator.

2. Unscrew the leads going to the capacitors. Check the H771 prints to ensure that the wiring matches the prints.

3. Remove the plugs from the M7745 and M7744.

4. Remove the six screws holding the regulator to the power supply chassis.

5. Replace the regulator by performing steps 1–4 in reverse.

6.5.2 Drive Replacement Procedure

1. With power removed, remove the plenum and power plug from the rear of the drive (Figure 6-1).
2. Raise the M7744 module to the service position.
3. Remove the plugs for this drive (Table 6-3).
4. Loosen the six screws securing the drive to the chassis.
5. While holding the drive, remove the screws from the four corners.
6. Carefully remove the remaining screws without allowing the drive to drop down.
7. Slowly lower the drive, guiding the wiring as the drive is lowered.
8. To install a drive, place the two center screws in the holes in the chassis.
9. Raise the drive, guiding the wiring through the hole.
10. With the drive centered, start the two screws carefully so as not to cross-thread them. Do not tighten these screws all the way.
11. Start the remaining screws, being careful not to cross-thread them.
12. Tighten all screws including the screws for the plenum.
13. Insert the plugs listed in Table 6-3.
14. Insert the power plug.
15. Secure the M7744 module.

6.5.3 Front Handle Replacement Procedure

1. Remove the drive carefully as described in Paragraph 6.5.2.
2. While holding the cone, lift the cover into place and remove the two screws securing the handle to the cover.
3. Remove the handle by pulling it through the openings in the bezel.
4. Replace the handle but do not tighten the screws all the way.
5. Adjust the handle so that the handle adapters do not interfere with the bezel. Latch the handle properly.
6. Tighten the screws.
7. Recheck for interference and replace the drive.

6.5.4 Drive Motor Replacement Procedures

1. Remove the drive as described in Paragraph 6.4.2.
2. Remove the bottom cover.
3. Disconnect the motor wires.
4. Remove the drive belt.
5. Remove the pulley from the motor shaft.
6. Remove the nuts securing the motor to the chassis and remove the motor.
7. Insert the new motor and secure it loosely in place.
8. Replace the pulley on the new motor shaft.
9. Adjust the height of the motor pulley until the two pulleys are in line. (Pulley may be adjusted by placing a straight edge such as a scale between the two pulleys.)
10. Place the DEC tool (part #9306353-3 for 50 Hz motor or 9306353-2 for 60 Hz motor) between the spindle pulley and motor pulley to measure the correct distance between both pulleys and then secure the motor in place (Figure 6-2).
11. Place the drive belt on the pulleys.
12. Replace the drive bottom cover and then reinstall the drive.

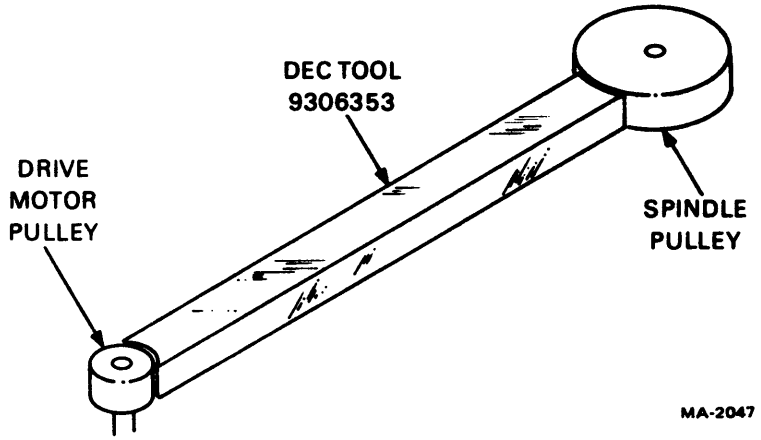


Figure 6-2 Drive Motor Positioning Diagram

6.5.5 Drive Belt Replacement Procedures

1. Discard the old belt.
2. Clean the pulleys and motor shaft with freon, 90% alcohol, or a suitable non-oil base solvent.
3. Loosen the drive motor nuts.
4. Place the belt spacer tool (9306353-2 for 60 Hz motor, or 9306353-3 for 50 Hz motor) between the pulleys (motor and spindle) and move the motor so that the tool is snug (Figure 6-2).
5. Tighten the motor nuts to secure the motor.
6. First loop the belt over the motor pulley; then loop the belt over the spindle pulley.
7. The belt will now be mounted with the correct tension.

NOTE

The belt used is made of nylon, not rubber, and it will not exert more tension if it is stretched beyond the diameter set by the tool. If the belt is stretched beyond the distance set by the belt spacer tool, it will be damaged because of the belt construction.

6.5.6 Quick Check For Belt on Pulleys

The RX02 belt is easily checked by sliding it forward on slides and viewing the belt path through the opening between the sheet metal chassis and the bottom cover.

APPENDIX A
RX02 SUMMARY

Interface	Density	Computer
RX8E-M8357	Single	PDP-8 - Prog I/O
RX28-M8357	Single or double	PDP-8 - Prog I/O
RX11-M7846	Single	PDP-11 - Prog I/O
RX211-M8256	Single or double	PDP-11 - DMA
RXV11-M7946	Single	LSI-11 - Prog I/O
RXV21-M8029	Single or double	LSI-11 - DMA

(Address for RX2C5 = 177170, for RX2E5 = 177172)

Controller M7744

Interface

RX8E/RX11/RXV11
RX211/RXV21
RX28

Switch Settings

S1-1	S1-2	
ON	OFF	
OFF	ON	
OFF	OFF	
ON	ON	←Illegal settings

Read/Write Electronics M7745

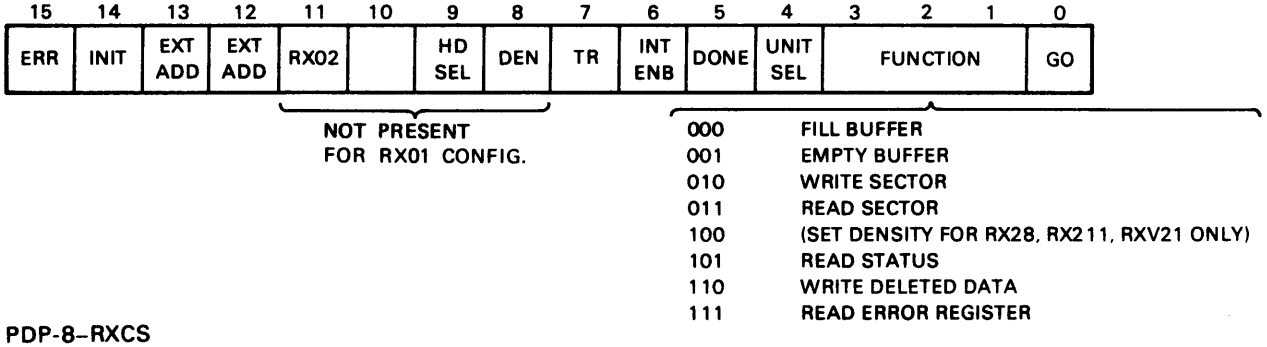
Power supply
H771A: 115 V/60 Hz,
H771C: 115 V/50 Hz,
H771D: 230 V/50 Hz

Left Drive = Drive 0
Right Drive = Drive 1

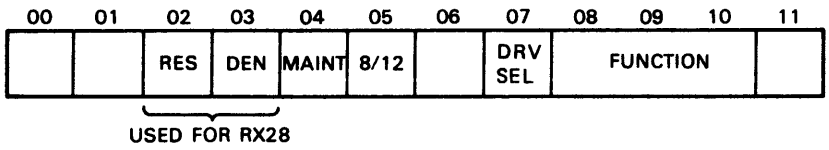
Single Drive 60 Hz = RX02-CA
Single Drive 50 Hz = RX02-CC

Track Address: $0–114_8$ (0–76)
Sector Address: $1–32_8$ (1–26)

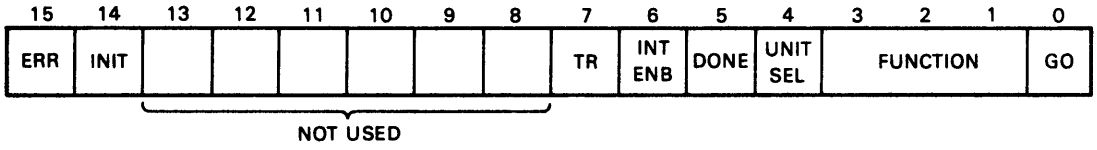
RX2CS



PDP-8-RXCS

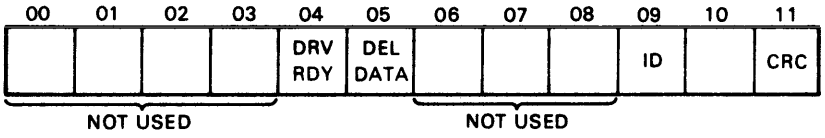


PDP-11/LSI-11-RXCS

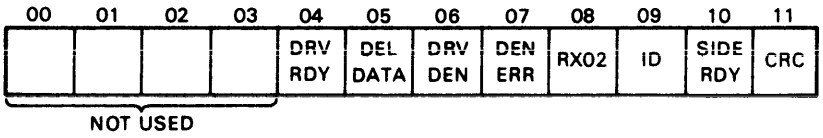


MA-2050

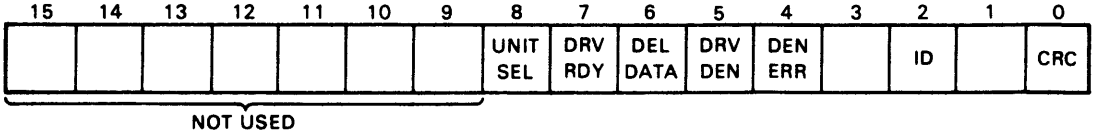
RX8E RXES



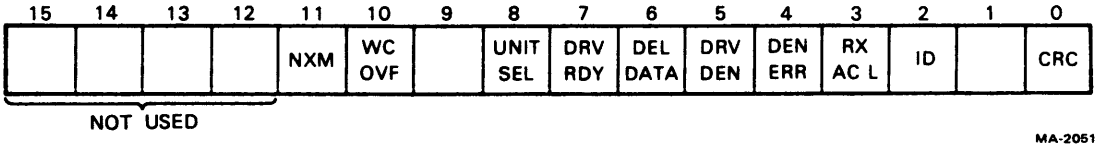
RX28 RX2ES



RXII/RXVII RXES



RX211/RXV21 RX2ES



MA-2051

Temperature

RX02	nonoperating: operating:	-35° to +60° C (-30° to +140° F) 15° to 32° C (50° to 90° F)
media	nonoperating:	-35° to 52° C (-30° to 125° F)

Relative Humidity

RX02	operating:	25° C (77° F) max wet bulb 2° C (36° F) min dew point 20% to 80% rel humidity
	nonoperating:	5% to 98% rel humidity (no condensation)
media	nonoperating:	10% to 80% rel humidity

Electrical

AC power input	4 A at 115 Vac 2 A at 230 Vac
RX02	5 A @ +5 Vdc, 25 W; 0.14 A @ -5 Vdc, 0.7 W; 1.3 A @ +24 Vdc, 31 W
Interface module	
M7946, M8029	1.8 A @ +5 Vdc
M7846, M8256	1.8 A @ +5 Vdc
M8357	1.5 A @ +5 Vdc

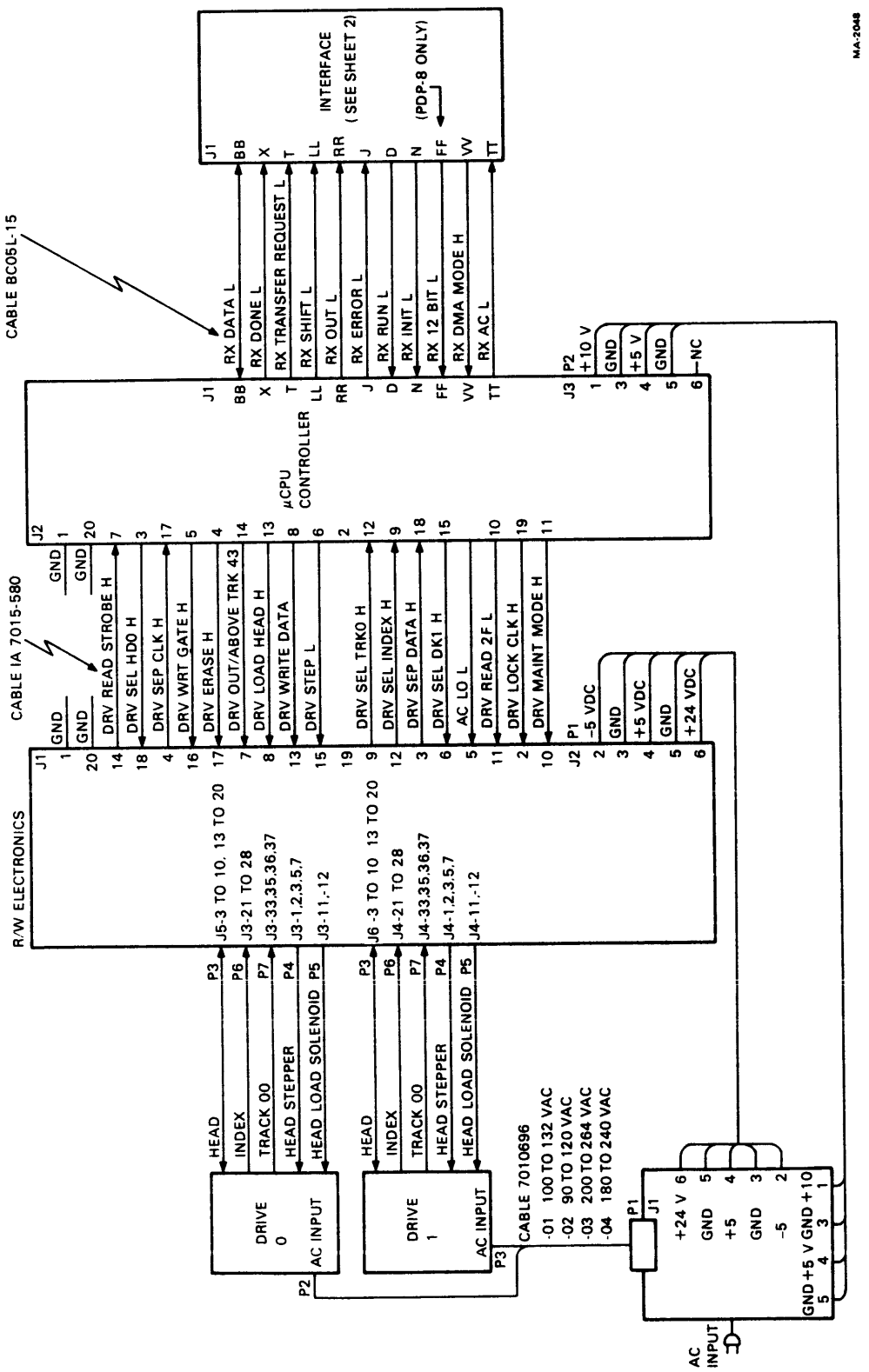


Figure A-1 RX02 System Interconnection Diagram (Sheet 1 of 2)

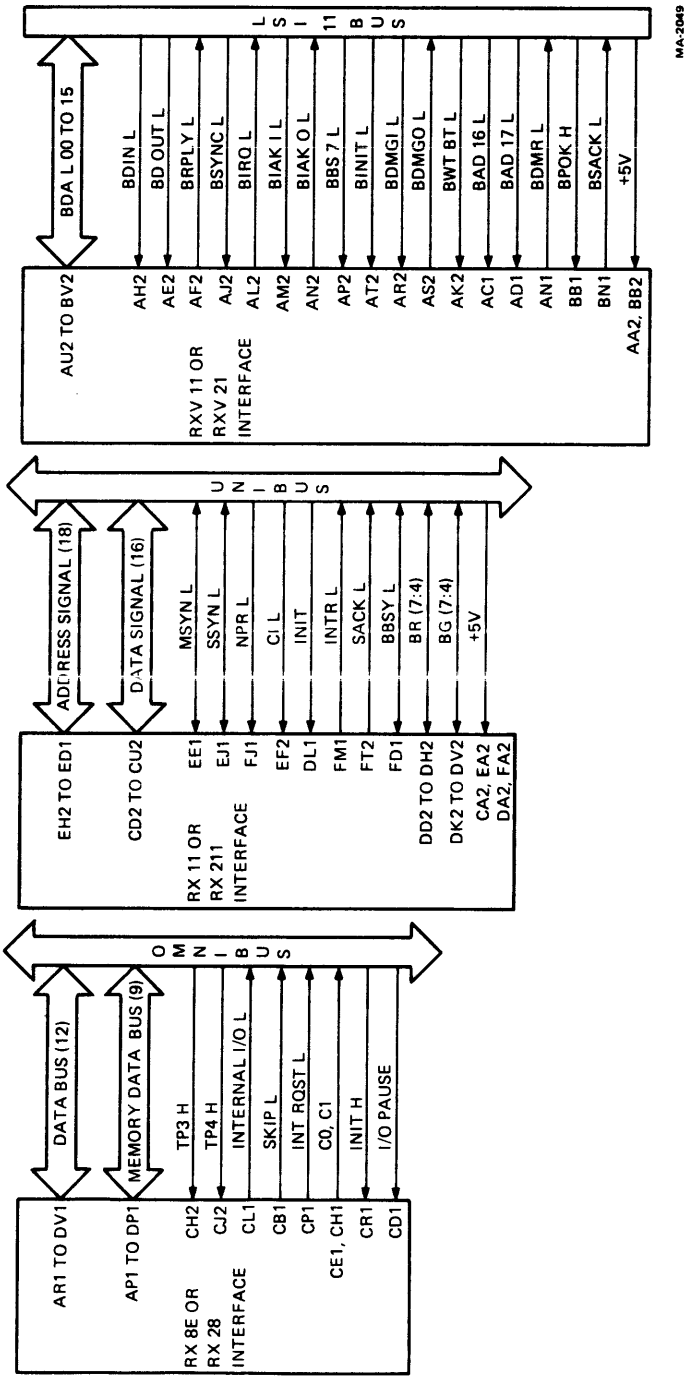


Figure A-1 RX02 System Interconnection Diagram (Sheet 2 of 2)

— — — — — — — — — — — — **Fold Here** — — — — — — — — — — — —

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

FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.

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

Postage will be paid by:

Digital Equipment Corporation
Technical Documentation Department
Maynard, Massachusetts 01754

RX02 FLOPPY DISK SYSTEM
TECHNICAL MANUAL
EK-0RX02-TM-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-0RX02-TM-001