EK-RUX50-TM-001

RUX50 Controller

Technical Manual

digital

EK-RUX50-TM-001

RUX50 Controller

Technical Manual

Prepared by Educational Services
of
Digital Equipment Corporation

First Edition, October 1984

Copyright © 1984 by Digital Equipment Corporation.
All Rights Reserved.

Printed in U.S.A.

The reproduction of this material, in part or whole, is strictly prohibited. For copy information, contact the Educational Services Department, Digital Equipment Corporation, Maynard, Massachusetts Ø1754.

The information in this document is subject to change without notice. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

IBM is a trademark of International Business Machines Corporation.

The following are trademarks of Digital Equipment Corporation, Maynard, Massachusetts.

digital	DIGITAL	RSTS
DECmate	MASSBUS	RSX
DECnet	Micro/PDP-11	UNIBUS
DECsystem-1Ø	PDP	VAX
DECSYSTEM-2Ø	P/OS	VMS
DECUS	Professional	VT
DECwriter	Q-Bus	Work Processor
DIBOL	Rainbow	

CONTENTS

TABLES

CHAPTER 1
INTRODUCTION

1.1 GENERAL INFORMATION

This manual describes the configurations, installation, and operation of the RUX5Ø controller module as a diskette drive subsystem. The manual identifies associated diskette drives, but it does not describe them in detail.

- Chapter 1 briefly describes the RUX5Ø and presents its features.
- Chapter 2 tells how to configure and install the RUX5Ø as a subsystem.
- Chapter 3 describes the RUX5Ø's operation.
- Chapter 4 provides detailed information on the RUX5Ø connectors.
- Chapter 5 covers troubleshooting. It explains how to interpret errors detected by the RUX5Ø's self-test and diagnostics.
- Chapter 6 describes the programmable registers that are UNIBUS addressable on the RUX5Ø. It briefly covers mass storage control protocol (MSCP) and diagnostics and utilities protocol (DUP). To give you a better understanding of the RUX5Ø's operation, Chapter 6 also describes the internal hardware registers.
- Chapter 7 lists environmental and functional specifications for the RUX5Ø as well as data formats and supported drive characteristics.

1.2 RELATED DOCUMENTATION

The following documents provide more information on the RUX50.

Title	Number
RUX50 Field Maintenance Print Set	MP002006
UDA50 Programmer's Documentation Kit	QP-905-GZ
RX50-D, -R Dual Flexible Disk Drive Subsystem Owner's Manual	EK-LEP01-OM

For extra information, refer to documentation for the system in which the RUX50 is installed.

1.3 GENERAL DESCRIPTION

The RUX50 (M7522) is a UNIBUS controller for 5-1/4-inch diskette drives. It is a quad-size, single-board controller that recognizes a subset of MSCP.

The RUX50 interfaces RX50 diskette drives to any hex-size backplane that implements an 18-bit/16-bit UNIBUS. It also supports other drives:

Single-density, single-sided, dual diskette drives
Single-density, double-sided, single diskette drives
Double-density, double-sided, single diskette drives.

1.4 FEATURES

The RUX50 has the following features.

- Is a single, quad-size module.
- Supports direct memory access (DMA) data transfers in 18-bit/16-bit addressing modes.
- Supports 18-bit addressing on a UNIBUS.
- Interfaces up to four logical units.
- Implements drives other than Digital Equipment Corporation's.
- Is compatible with Micro/PDP-11 and Professional RX50 diskettes.

CHAPTER 2
CONFIGURATION AND INSTALLATION

2.1 INTRODUCTION

The RUX5Ø is a direct memory access (DMA) device that mounts in any small peripheral controller (SPC) slot of the UNIBUS backplane. A single RUX5Ø interfaces up to four logical units to the host processor. The RUX5Ø must conform to valid subsystem (Paragraph 2.2), jumper (Paragraph 2.3), and cabling configurations (Paragraph 2.4).

The RX5Ø diskette drive is a random access storage device that uses two single-sided, 133.4-mm (5-1/4-in) RX5ØK diskettes. The total storage capacity of the RX5Ø is 8ØØ kilobytes of formatted data.

2.2 SUBSYSTEM CONFIGURATIONS

Typically, the RUX5Ø interfaces one or two RX5Øs to the host system. Each RX5Ø is two logical units. The device type jumpers (Paragraph 2.3) must be set for each logical unit attached to the RUX5Ø. Table 2-1 lists acceptable RUX5Ø subsystems. Dual diskette drives must start on the first and third logical unit boundaries assigned to the RUX5Ø (Paragraph 2.3.2).

Table 2-1 Subsystem Configurations

Configuration	Number and Type of Diskette Drives	Logical Diskette Drive Numbers
1	One RX5Ø	Units Ø, 1 = RX5Ø
2	Two RX5Øs	Units Ø, 1 = RX5Ø Units 2, 3 = RX5Ø
3	One single logical unit drive	Unit Ø = single unit
4	One RX5Ø and one single logical unit drive	Units Ø, 1 = RX5Ø Unit 2 = single unit

2.3 JUMPER CONFIGURATIONS

To change the RUX50's device address, logical unit number, device type, logical location, and long- or short-line transmission from their standard settings, reconfigure the jumpers on the RUX50. Figure 2-1 shows the jumper locations.

2.3.1 Device Address Selection

Figure 2-1 shows the location of the RUX50 address jumpers. Table 2-2 lists the jumper configuration for the standard address, 772150. To configure the RUX50 for an address other than 772150, use Table 2-2 to determine the appropriate configuration for address bits A2 through A13 on jumpers W4 through W15.

Table 2-2 Device Address Jumper Configuration

Jumper	UNIBUS Address Bit	Standard 772150 State	Standard Binary State	Standard Octal State
W4	A2	in	0	0
W5	A3	out	1	
W6	A4	in	0	5
W7	A5	out	1	
W8	A6	out	1	
W9	A7	in	0	1
W10	A8	in	0	
W11	A9	in	0	
W12	A10	out	1	2
W13	A11	in	0	
W14	A12	out	1	7
W15	A13	out	1	

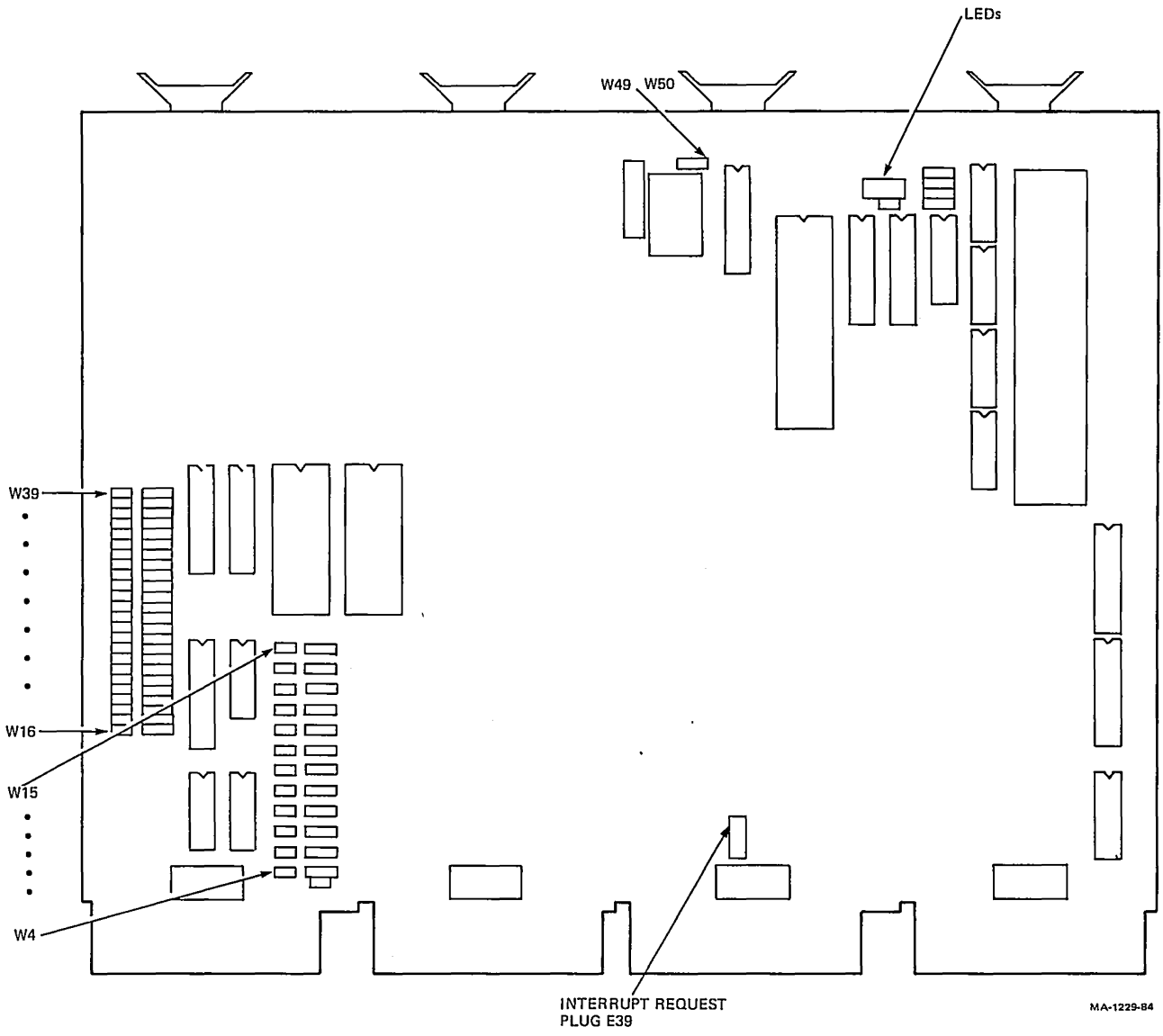


Figure 2-1 RUX5Ø Jumper Locations

2.3.2 Logical Unit Number Selection

Figure 2-1 shows the location of the RUX5Ø's logical unit number (LUN) jumpers. Set these jumpers to the lowest LUN assigned to any diskette drive controlled by the RUX5Ø. The standard configuration for the LUN jumpers is Ø. Use Table 2-3 to configure the RUX5Ø for its LUN.

2.3.3 Device Type Selection

Sixteen jumpers identify each device connected to the RUX5Ø. These jumpers are standardly configured for the RX5Ø. They are organized in four groups of four jumpers. The RUX5Ø supports four logical units, so each jumper group defines the device for each logical unit. Dual logical units must occupy the first or third jumper group. Single logical units can occupy any jumper group as long as the dual logical units have the first or third. Table 2-4 defines the jumper states for various drive types. (Refer to the appendix for drive type descriptions.)

2.3.4 Interrupt Request Level

The RUX5Ø interrupts at priority level 5, which is determined by the interrupt request plug in E39. Figure 2-1 shows E39 on the module.

2.3.5 Long- or Short-Line Selection

The RUX5Ø supports drive cable configurations that are long and short. A long-line configuration (standard) allows 4Ø ft (maximum) or less of cable between the RUX5Ø and the drive. A short-line configuration allows 1Ø ft (maximum) or less of cable between the RUX5Ø and the drive.

Install jumper W49 for long-line configurations and jumper W5Ø for short-line configurations.

> **NOTE**
> **For most applications, the long-line configuration is typical. If the total cable length between the RUX5Ø and drive is less than 1Ø ft, you can use the short-line configuration.**

Table 2-3 Logical Unit Number Jumper Configuration

First Logical Unit	Jumper States W32	W33	W34	W35	W36	W37	W38	W39
Ø	out	out	out	out	out	out	out	out
1	out	out	out	out	out	out	out	in
2	out	out	out	out	out	out	in	out
3	out	out	out	out	out	out	in	in
4	out	out	out	out	out	in	out	out
5	out	out	out	out	out	in	out	in
6	out	out	out	out	out	in	in	out
7	out	out	out	out	out	in	in	in
8	out	out	out	out	in	out	out	out
...								
255	in	in	in	in	in	in	in	in

Table 2-4 Device Type Jumper Configuration

Jumper	Bit	Logical Unit Location	Jumper States for Device Type 1	2	3	4
W31	Ø	First	in	out	in	out
W3Ø	1	First	out	in	in	out
W29	2	First	out	out	out	in
W28	3	First	out	out	out	out
W27	Ø	Second	in	out	in	out
W26	1	Second	out	in	in	out
W25	2	Second	out	out	out	in
W24	3	Second	out	out	out	out
W23	Ø	Third	in	out	in	out
W22	1	Third	out	in	in	out
W21	2	Third	out	out	out	in
W2Ø	3	Third	out	out	out	out
W19	Ø	Fourth	in	out	in	out
W18	1	Fourth	out	in	in	out
W17	2	Fourth	out	out	out	in
W16	3	Fourth	out	out	out	out

Legend
1 Single-sided, dual drive or RX5Ø (96 tracks per inch)
2 Single-sided, dual drive (48 TPI)
3 Double-sided, single drive (96 TPI)
4 Double-sided, single drive (48 TPI)

2.4 SUBSYSTEM INSTALLATION

Install the RUX5Ø (M7522) in any slot of the UNIBUS backplane. Follow these rules when installing the RUX5Ø into the UNIBUS backplane.

- After installing the RUX5Ø, make sure all empty slots have a grant card (PN G727 or G7273).
- Remove the nonprocessor request (NPR) grant jumper for the slot containing the RUX5Ø from the backplane.

Before installing the RUX5Ø, make sure its jumpers for device address, logical unit number, device type, logical location, and long- or short-line transmission are configured correctly (Paragraph 2.3).

The RUX5Ø subsystem consists of the controller module, the signal cable to the interconnect panel, and the long-line adapter that installs in the diskette drive. Figure 2-2 shows a typical subsystem.

Connect the 5Ø-conductor signal cable (PN 17-ØØ743-Ø6) to the J1 connector on the RUX5Ø. Connect the other end of this cable to the signal distribution panel in the host cabinet.

NOTE

FCC cabinets already have the mounting bracket and interconnect panel. For other cabinets, install a mounting bracket (PN 74-27292) if necessary.

For all long-line configurations, install the long-line adapter in the diskette drive (Figure 2-3). The adapter goes between the drive's interconnect panel and flat ribbon cable. Four jumpers are on the long-line adapter (W1, W2, W3, and W4). Jumper W1 is always removed for RX5Øs. Jumpers W2, W3, and W4 are installed when the diskette drive is the last unit or a single unit for daisychained diskette drives. Remove all the jumpers if the drive is the center unit in a daisychain configuration.

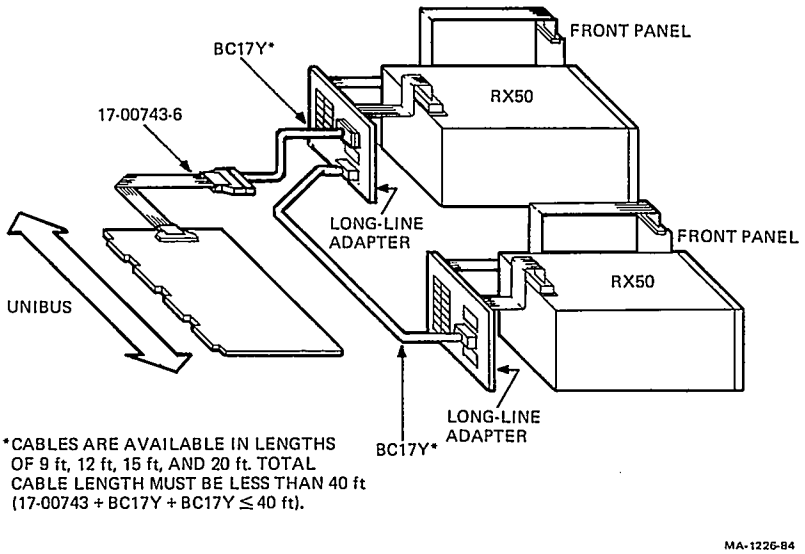


Figure 2-2 RUX50 Subsystem Installation

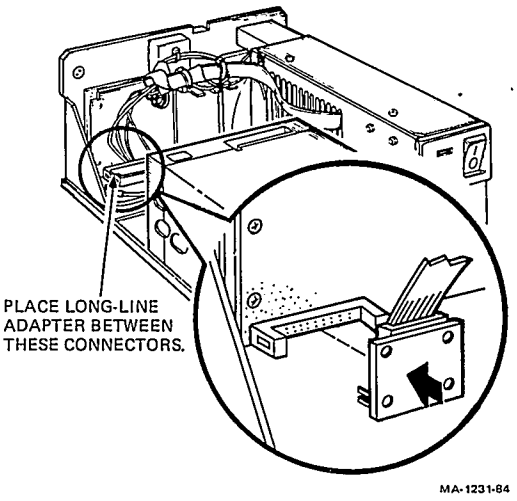


Figure 2-3 Long-Line Adapter Installation

CHAPTER 3
THEORY OF OPERATION

3.1 INTRODUCTION

This chapter gives a general description of how the circuits on the RUX5Ø function. Then it gives a detailed description of each circuit.

3.2 CIRCUIT OVERVIEW

The RUX5Ø has four major circuit functions that interface 5-1/4-inch diskette drives to the UNIBUS. These circuits interact to store and retrieve data between the UNIBUS controller (host processor) and the diskette drives. Figure 3-1 shows these circuits.

UNIBUS interface circuits
T-11 processor circuits
DMA control circuits
Diskette drive controller circuits

3.2.1 UNIBUS Interface Circuits

These circuits (Figure 3-1) contain the MSCP registers and the UNIBUS interfacing circuits required by a SPC UNIBUS device. They perform the following standard UNIBUS operations.

- Initialize the entire RUX5Ø via an initialization and poll (IP) write.
- Initiate the RUX5Ø to start polling via a IP read.
- Read and write the status and address (SA) register from the UNIBUS.
- Enable the vector address onto the UNIBUS data lines.
- Perform DMA request handshaking between the UNIBUS and DMA control circuits.

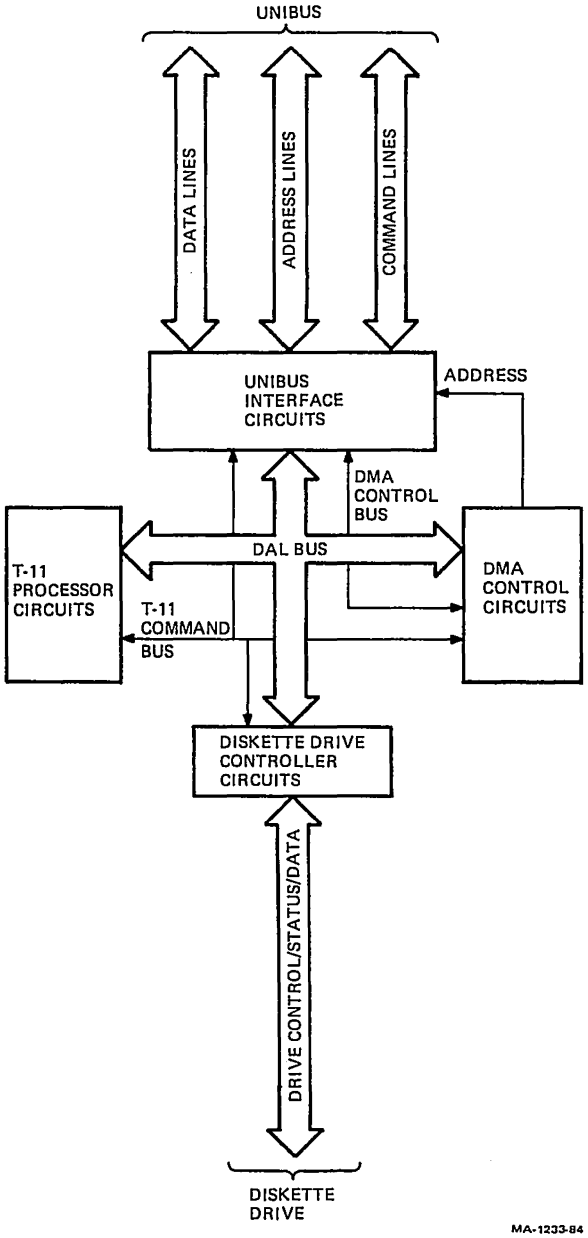


Figure 3-1 RUX50 Circuits

3.2.2 T-11 Processor Circuits

These circuits (Figure 3-1) control the operation of the RUX50. They contain the microcode that interprets and responds to the MSCP accesses that the RUX50 receives from the UNIBUS interface circuits. The T-11 processor circuits perform the following standard operations.

- Read and write the SA register from T-11 memory.
- Load the vector register in the UNIBUS interface circuits from T-11 memory.
- Initialize the RUX50 when the UNIBUS interface circuits detect a IP write. This function includes the following operations.

  T-11 runs preliminary diagnostics.

  T-11 reads the SA write register for the length of the MSCP command/response rings and the interrupt enabling, and places this data in the RUX50's RAM.

  T-11 writes the SA read register with its command and ring lengths.

  T-11 then reads the SA write register for ring base low and ring base high starting addresses.
- For DMA accesses, the T-11 processor circuits move data between the DMA data buffer register in the UNIBUS interface circuits and T-11 memory.
- T-11 reads and writes the drive control and status register (CSR) in the diskette drive controller circuits.
- T-11 reads and writes between the diskette drive controller chip (WD1793) and the T-11 or T-11 memory.

3.2.3 DMA Control Circuits

These circuits (Figure 3-1) interact with the UNIBUS interface and T-11 processor circuits to perform DMA transfers between the host's main memory and T-11 memory. To perform this function, the DMA control circuits do the following operations.

- Initiate and monitor DMA transfer requests to the UNIBUS interface circuits.
- Generate bus and data transfer synchronization signals for the UNIBUS interface circuits.

- Provide DMA status information on request to the T-11 processor circuits.
- Provide memory address for DMA transfer to the UNIBUS interface circuits.

3.2.4 Diskette Drive Controller Circuits

The T-11 processor circuits control the diskette drive controller circuits (Figure 3-1) to access the diskette drives. To perform this function, these circuits do the following operations.

- Receive control information from the T-11 for the drives.
- Receive status information from the drives for the T-11.
- Convert data in T-11 memory to diskette drive formatted data and send it to the drive.
- Retrieve diskette drive formatted data from the drive and convert it to 8-bit words for T-11 memory.

3.3 CIRCUIT DETAILS

This section describes the RUX5Ø's basic architecture and data path relationships for each major circuit function. Figure 3-1 shows that the RUX5Ø is a bus-oriented subsystem. The host system controls the subsystem, and the T-11 processor and DMA control circuits share it.

3.3.1 UNIBUS Interface Circuits

These circuits (Figure 3-2) consist of the following components.

UNIBUS address transceivers/address match detector
UNIBUS data buffer/transceivers
UNIBUS control transceivers
Port input/output (PIO) read/write SA registers
PIO controller/IP register
DMA request controller
Interrupt controller
DMA data buffer/transceivers
Vector register

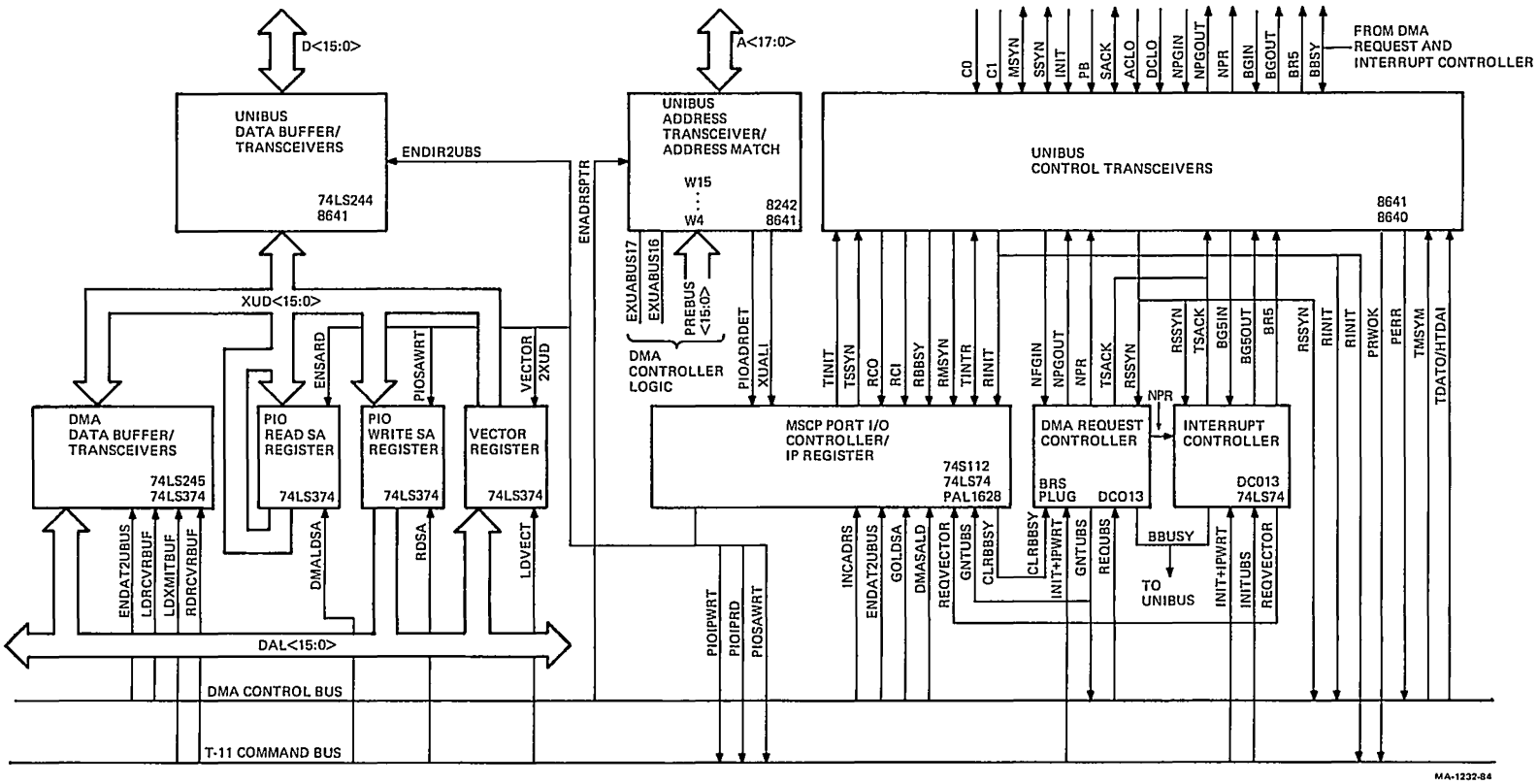


Figure 3-2 UNIBUS Interface Circuit

3.3.1.1 UNIBUS Address Transceivers/Address Match Detector -- This circuit compares the jumpered address to the UNIBUS address. When an address match occurs (indicating an SA or IP read/write), this circuit asserts PIOADRDET. PIOADRDET alerts the PIO controller to enable the appropriate data paths and interrupts in the RUX5Ø.

This circuit also receives the UNIBUS DMA address (PREBUS<15:Ø>) and UNIBUS address extension (EXUABUS<17:16>) from the DMA control circuits. The DMA control signal ENADRSPTR enables this address onto the UNIBUS address lines.

3.3.1.2 UNIBUS Data Buffer/Transceivers -- This circuit is 16 bits wide and passes data between the UNIBUS and the XUD bus. It is enabled by the DMA controller (through the pio controller) or by the PIO controller during MSCP accesses.

3.3.1.3 UNIBUS Control Transceivers -- The UNIBUS control transceivers buffer and transfer the UNIBUS control signals between the UNIBUS controller (host processor) and the RUX5Ø.

3.3.1.4 PIO Read/Write SA Registers -- These SA registers are used by the T-11 and the UNIBUS processor (host computer) for the MSCP port initialization sequence. These registers occupy the same UNIBUS I/O page address. The SA write register is written by the UNIBUS processor and read by the T-11. The SA read register is written by the T-11 and read by the UNIBUS processor. During the initialization sequence, the UNIBUS controller writes the MSCP command buffer address and interrupt vector to the SA write register for the T-11. The T-11 writes the SA read register with the initialization status data for the UNIBUS processor. The SA read and write registers are both 16 bits wide.

3.3.1.5 MSCP PIO Controller/IP Register -- This circuit performs the following functions simultaneously.

- It detects MSYN and sends SSYN to or from the UNIBUS control transceivers after the host processor receives or sends data.
- The PIO control logic also enables output and direction of the UNIBUS data transceiver.
- The PIO controller decodes the type of transfer by the CØ and C1 signals for an IP read, IP write, SA read, or SA write.

- After the UNIBUS address transceivers/address match detector detects a valid address, the PIO controller/IP register loads the SA write register from the UNIBUS or enables the SA read register to the UNIBUS.

After receiving a request signal from the interrupt controller (REQVECTOR), the PIO controller gates the vector from the vector register onto the UNIBUS through the UNIBUS data buffer/transceivers and the UNIBUS interrupt signal (TINTR) through the UNIBUS control transceivers. The PIO controller sends interrupt signals (PIOIPWRT, PIOIPRD, PIOSAWRT) to the T-11 processor circuits whenever the host processor reads or writes the SA or IP registers.

3.3.1.6 Interrupt Controller and Vector Register -- The interrupt vector register contains the vector that is gated to the UNIBUS during an interrupt sequence. The MSCP initialization sequence supplies the T-11 with the vector. The T-11 then loads the interrupt vector register. The contents of this register transfers to the UNIBUS under control of the interrupt controller (via the PIO controller). The vector register is 8-bits wide and ranges in value from Ø to 774. The normal interrupt vector is 154 and the priority level is 5.

3.3.2 T-11 Processor Circuits

These circuits (Figure 3-3) consist of the following components.

Interrupt, request, initialize, and asynchronous hold registers
T-11 microprocessor
Mode buffer
Address latch
I/O bit map and memory decode
Random access memory (RAM)
Read-only memory (ROM)
Display register and light emitting diodes (LEDs)
Device register
LUN register

3.3.2.1 Interrupt Request, Initialize, and Asynchronous Hold Registers -- The interrupt request register performs conditional priority on interrupt requests to the T-11. It stores the results in the asynchronous holding register (AHR) every T-11 column address strobe (CAS) cycle. The output of the AHR goes directly into the T-11 interrupt port.

The initialize register monitors the UNIBUS AC LO and DC LO (PWROK) signals to determine when to perform power-up or power-down functions. The register also begins an initialization sequence when the host processor writes to the IP register or during UNIBUS initialization.

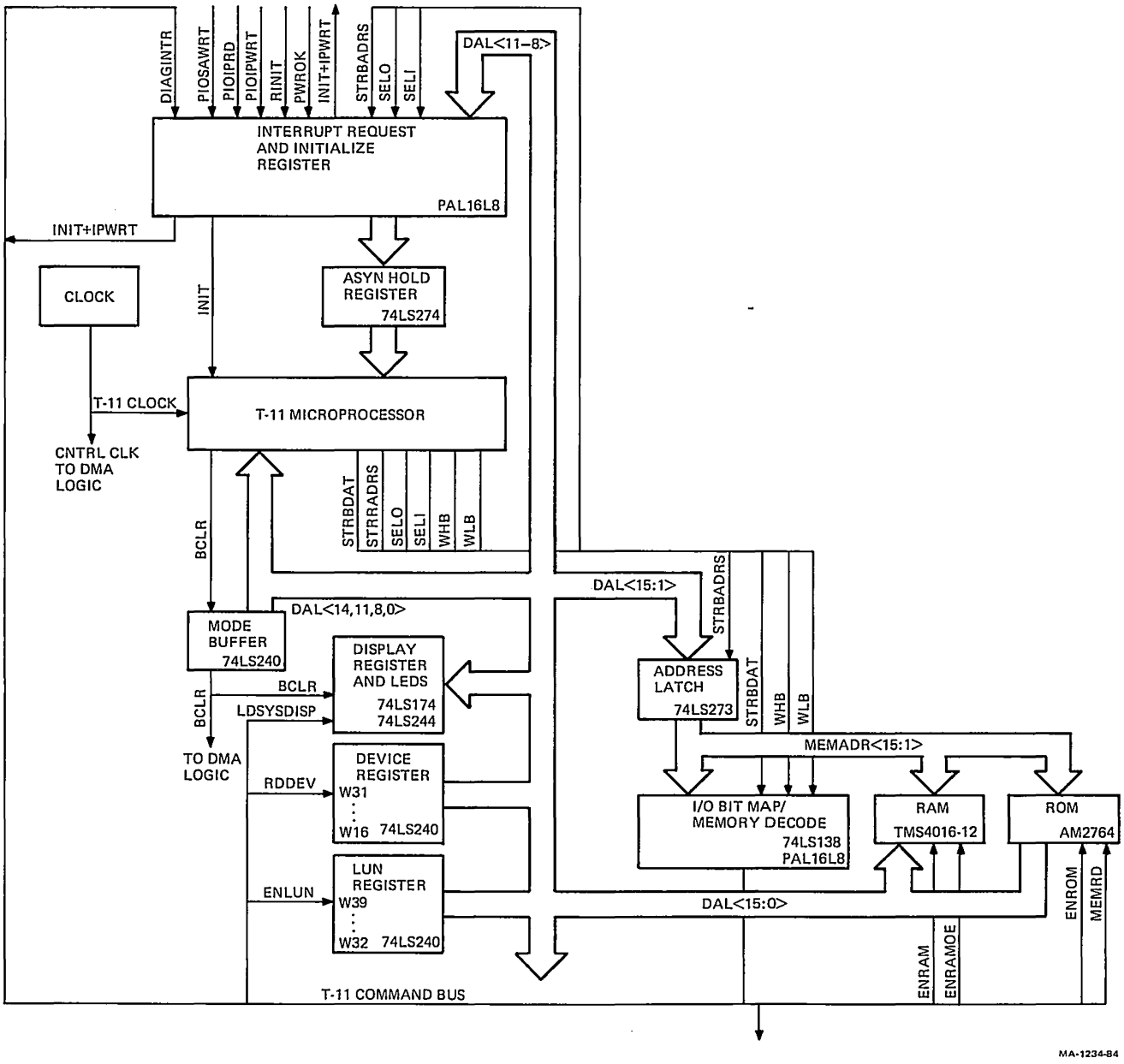


Figure 3-3 T-11 Processor Circuit

3.3.2.2 T-11 Microprocessor, Mode Buffer, Address Latch, I/O Bit Map, and Memory Decoder -- The supervisory control of the RUX5Ø is done by the T-11 with an input clock frequency of 7.5 MHz. During a power-up sequence, the initialize register (Paragraph 3.3.2.1) configures the T-11 for the 16-bit mode and enables the T-11 to use static RAM.

The address latch latches the T-11 memory address onto the memory address lines (MAL) during the address phase of the bus cycle. This allows the T-11 to point to any specific location in the ROM, RAM, or I/O bit map memory decoder. The I/O bit map and memory decoder enables and loads all the registers and devices that the T-11 can access.

The T-11 performs the following operations.

1. Sends the host processor status information through the DMA data buffer register. The DMA controller moves data from the DMA data buffer to the SA read register. If prior to the T-11 loading the DMA data buffer register, the T-11 sets the GOLDSA flag in the DMA data buffer.

2. Reads the host processor's status information through the SA write register.

3. After receiving the host vector address from the SA write register during port initialization, the T-11 loads the vector address into the vector register.

4. The T-11 initiates an interrupt to the host processor when the T-11 finishes performing the instructions in the host command ring in T-11 RAM. This happens if interrupts were enabled during port initialization.

5. Reads and writes the drive CSR and reads the LUN and device type register.

6. Reads and writes all data from or to the diskette drive controller chip from T-11 RAM.

7. Interaction between the T-11 and the DMA control circuits brings host command rings and data to be written to the drives into T-11 RAM.

    a. The T-11 loads the source UNIBUS address into the UNIBUS address registers. The DMA state machine increments these registers after each word transfer.

    b. The T-11 then polls the DMADONE bit in the DMA data buffer CSR (DDBCSR).

c. If the DMADONE bit is set, the T-11 sets the GODMAFUBS bit in the DDBCSR, which initiates the DMA control circuits to get a word from host memory referenced from the UNIBUS DMA address pointer.

d. The T-11 then polls the RCVRFUL bit in the DDBCSR. If this is set, the T-11 writes the contents of the DMA data buffer register into the T-11 RAM.

e. The T-11 then checks its internal word counter to determine if it should repeat the process. If the internal word counter is not equal to the number of words of a particular transfer, then the T-11 increments its internal T-11 word counter and repeats steps b through e until the end of a particular transfer.

8. Interaction between the T-11 and the DMA control circuits brings RUX50 response rings and data from the drives to T-11 RAM for transfer to host memory.

a. The T-11 loads the destination UNIBUS address into the UNIBUS address registers. The DMA state machine increments these registers after each word transfer.

b. The T-11 then polls the XMITFUL bit in the DMA data buffer CSR (DDBCSR).

c. If the DMA data buffer is empty (XMITFUL = 0), then the T-11 loads the DMA data buffer register with the data to be transferred to the host memory from T-11 RAM. The loading of the DMA data buffer initiates the DMA controller to transfer the word from the DMA data buffer register to the host memory referenced from the UNIBUS DMA address pointer.

d. The T-11 then checks its internal word counter to determine if it should repeat the process. If the internal word counter is not equal to the number of words of a particular transfer, then the T-11 increments its internal T-11 word counter and repeats steps b through d until the end of a particular transfer.

3.3.2.3 RAM and ROM -- The 8K X 16 RAM has three major sections.

T-11 work space
Device control block
Sector buffer

The T-11 address space allocated to the RAM lies between 100000 and 120000 (octal). The 8K X 16 PROM just stores the T-11 instructions. Its T-11 address space lies between 000000 and 020000.

3.3.2.4 Display Register and LEDs -- T-11 processor circuits load the display register. This register controls the LEDs that show the RUX5Ø's status.

3.3.2.5 Device Type Register -- The device type register is a jumper-configured, 16-bit register. It indicates the type of drive for each of the four logical units attached to the RUX5Ø. (See Paragraphs 2.3.3 and 6.5.18 for details.)

3.3.2.6 LUN Register -- The LUN register is an 8-bit jumper-configured register. The T-11 reads this register. (See Paragraphs 2.3.2 and 6.5.17 for details.)

3.3.3 DMA Control Circuits Detail

The DMA control circuits (Figure 3-4) are made up of the following components.

DMA controller

Timers

Data buffer CSRs

UNIBUS DMA address pointer/buffer and UNIBUS address extension register

3.3.3.1 DMA Controller, Timers, and Data Buffer CSRs -- The DMA controller, when enabled, controls the UNIBUS DMA address pointer, UNIBUS address extention register, and the UNIBUS address and data transceivers. It also controls the loading of the SA read register, the DMA data buffer, and the DMA data buffer CSR. It operates with a 15-MHz clock input. In a DMA read, the source of the data is the DMA data buffer register and the destination of the data is the host memory. Similarly, in a write, the source of the data is the host memory and the destination of the data is the DMA data buffer.

The data transfer occurs as an NPR operation and obeys the UNIBUS DMA protocol. DMA transfers occur in one word transfers per bus acquisition. The DMA controller sets error flags and process complete flags in the DMA data buffer CSR after it does a DMA.

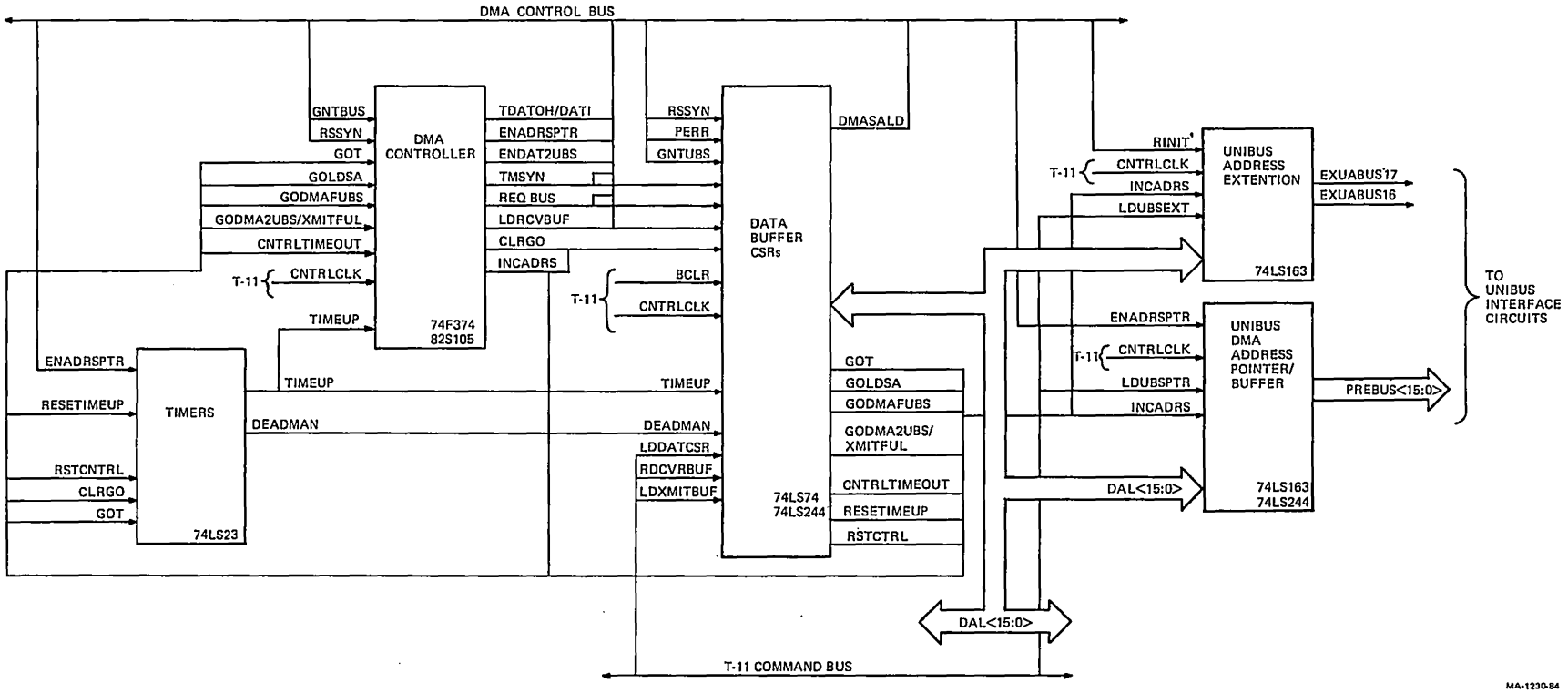


Figure 3-4 DMA Control Circuits

3.3.3.2 UNIBUS DMA Address Pointer/Buffer and UNIBUS Address Extension Register -- The UNIBUS DMA address pointer/buffer and UNIBUS address extension register consists of five loadable counters. The T-11 loads the beginning UNIBUS addresses into the counters. The T-11 loads the address extension before DMA operation if an extended address is necessary. The extended address is incremented if a transfer occurs across a 16-bit boundary. The output of these buffers goes to the UNIBUS interface circuit and is placed on the UNIBUS by the DMA controller.

3.3.4 Diskette Drive Controller Circuits

These circuits (Figure 3-5) are made up of the following components.

Diskette drive CSR
Data CSR
Diskette drive controller
Data separator/write precompensation
Drive transceivers

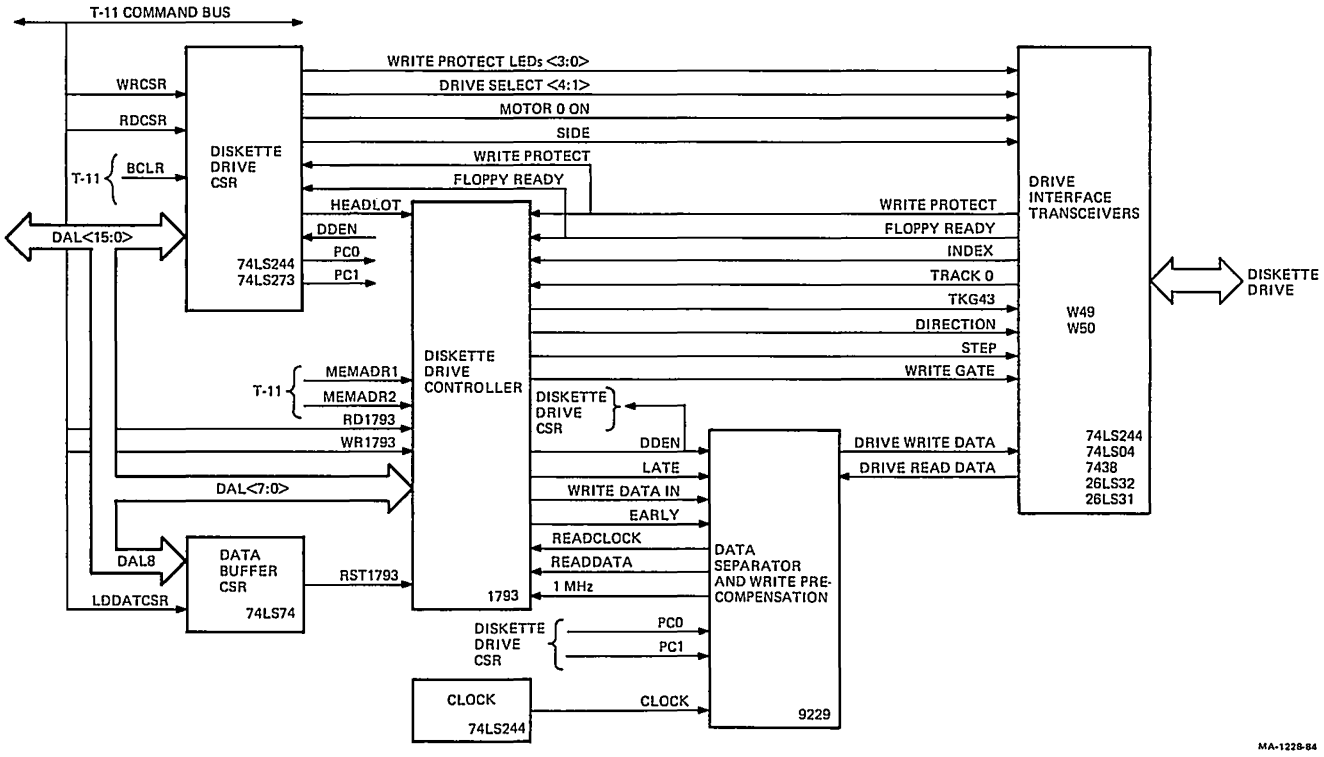


Figure 3-5 Diskette Drive Controller Circuit

3.3.4.1 Diskette Drive CSR and Data CSR -- The T-11 reads and writes the diskette drive CSR. The command register accepts a T-11 word that selects one of the four diskette drives and one of three precompensation values. This word also controls reads and writes in a single-density or double-density diskette drive format. It turns the drive motors on and lights the write protect LEDs on the drives.

NOTE

The T-11 turns all the drive motors off after no diskette activity has occurred for about 6 seconds or more to prolong the life of the drives.

The status register provides the T-11 with a data word that indicates the selected drive and the precompensation values. This word indicates whether the RUX5Ø reads and writes in a single-density or double-density diskette drive format. It shows the status of the drive motors and the write protect LEDs. It also shows whether the drive is ready and whether the selected diskette is write protected.

The T-11 processor circuits also load the data buffer CSR. This control register allows the T-11 to reset the diskette drive controller.

3.3.4.2 Diskette Drive Controller -- This controller (WD1793) performs all drive motor and data manipulations. It controls stepper motor direction. It also controls the conversion of data from serial to parallel and parallel to serial from the data separator/write precompensation circuit. This controller can read and write serial data in MFM (double density) or FM (single density) formats.

The T-11 loads the diskette drive controller with sector, track, data, and command information. The controller checks or generates a 16-bit cyclic redundancy check (CRC) from or to the selected drive.

3.3.4.3 Data Separator/Write Precompensation -- The data separator synchronizes the raw read signal from the selected drive and the free-running clock signal. It does this through a phase lock loop circuit, which captures the raw data directly from the drive within the pulse window of the clock.

The write precompensation circuit receives the serial data to be written to the drive and performs the following functions.

- Automatically generates a modified frequency modulated (MFM) bit stream.
- Automatically precompensates the MFM bit stream.

It has three possible precompensation values, depending on the position of the head. These precompensation values are selected from the diskette drive CSR. Depending on whether the early or late signals are generated from the WD1793, the write precompensation circuit adjusts the delivery timing of the serial data written to the drive. The operating frequency of the this circuit is 16 MHz.

3.3.4.4 Drive Interface Transceivers -- The drive interface transceiver communicates directly between the RUX5Ø and the diskette drives. Two jumpers allow the the drive transceivers to pass the read data from long-line or short-line ports on the connector to the data separator. The output and input of the transceiver go to a 5Ø-pin connector.

CHAPTER 4
INTERCONNECTIONS

4.1 INTRODUCTION

This chapter provides detailed descriptions of the UNIBUS connector (Paragraph 4.2) and the drive interface connector (Paragraph 4.3).

4.2 UNIBUS CONNECTOR

Table 4-1 lists the UNIBUS signal pin assignments that the RUX5Ø uses, and Table 4-2 defines the signals.

4.3 DRIVE INTERFACE CONNECTOR J1

The drive interface transceivers on the RUX5Ø connect to a 5Ø-pin connector (J1). The transceivers allow direct communication between the RUX5Ø and the diskette drives. The transceivers also provide open collector line driver output. Figure 4-1 shows the pins on connector J1. Table 4-3 lists pin signals and their functions.

Table 4-1 UNIBUS Signal Pin Assignments

Pin	Name	Pin	Name	Pin	Name	Pin	Name
CA1	NPG IN	CA2	+5V	EA1	--	EA2	+5V
CB1	NPG OUT	CB2	--	EB1	--	EB2	--
CC1	PA L	CC2	GND	EC1	A12 L	EC2	GND
CD1	--	CD2	D15 L	ED1	A17 L	ED2	A15 L
CE1	--	CE2	D14 L	EE1	MSYN L	EE2	A16 L
CF1	--	CF2	D13 L	EF1	AØ2 L	EF2	C1 L
CH1	D11 L	CH2	D12 L	EH1	AØ1 L	EH2	AØØ L
CJ1	--	CJ2	D1Ø L	EJ1	SSYN L	EJ2	CØ L
CK1	--	CK2	DØ9 L	EK1	A14 L	EK2	A13 L
CL1	--	CL2	DØ8 L	EL1	A11 L	EL2	--
CM1	--	CM2	DØ7 L	EM1	--	EM2	--
CN1	DC LO	CN2	DØ4 L	EN1	--	EN2	AØ8 L
CP1	--	CP2	DØ5 L	EP1	A1Ø L	EP2	AØ7 L
CR1	--	CR2	DØ1 L	ER1	AØ9 L	ER2	--
CS1	PB L	CS2	DØØ L	ES1	--	ES2	--
CT1	GND	CT2	DØ3 L	ET1	GND	ET2	--
CU1	--	CU2	DØ2 L	EU1	AØ6 L	EU2	AØ4 L
CV1	AC LO	CV2	DØ6 L	EV1	AØ5 L	EV2	AØ3 L
DA1	--	DA2	+5V	FA1	--	FA2	+5V
DB1	--	DB2	--	FB1	--	FB2	--
DC1	--	DC2	GND	FC1	SSYN L	FC2	--
DD1	--	DD2	BR7 L	FD1	BBSY L	FD2	--
DE1	--	DE2	BR6 L	FE1	--	FE2	--
DF1	--	DF2	BR5 L	FF1	--	FF2	--
DH1	--	DH2	BR4 L	FH1	--	FH2	--
DJ1	--	DJ2	--	FJ1	NPR L	FJ2	--
DK1	--	DK2	BG7 IN	FK1	--	FK2	--
DL1	INIT L	DL2	BG7 OUT	FL1	--	FL2	--
DM1	--	DM2	BG6 IN	FM1	INTR L	FM2	--
DN1	--	DN2	BG6 OUT	FN1	--	FN2	--
DP1	--	DP2	BG5 IN	FP1	--	FP2	--
DR1	--	DR2	BG5 OUT	FR1	--	FR2	--
DS1	--	DS2	BG4 IN	FS1	--	FS2	--
DT1	GND	DT2	BG4 OUT	FT1	GND	FT2	SACK L
DU1	--	DU2	--	FU1	--	FU2	--
DV1	--	DV2	--	FV1	--	FV2	--

Table 4-2 UNIBUS Signal Descriptions

Name	Mnemonic	Number of Lines	Function
Data Transfer Signals			
Address	A17:ØØ	18	The current bus master asserts the address lines. The host processor can address the RUX5Ø registers and the RUX5Ø can address memory locations during DMA data transfers.
Data lines	D15:ØØ	16	The data lines carry the word of information that is being transferred between the bus master and slave. Data lines carry commands, responses, DMA data, and interrupt vectors.
Control lines	C1:CØ	2	The bus master asserts the select lines and specifies the type of data transfer to take place. The KLESI-UA uses and recognizes three operations. **Name/C1:CØ/Function** Data in/ØØ/One word of data from slave to master Data out/1Ø/One word of data from master to slave Data out/11/One byte of data from master to slave
Master sync	MSYN	1	The bus master asserts and negates MSYN, which and is received by the bus slave. When MSYN is asserted, the master tells the slave to perform the function dictated by the C lines. When MSYN is negated, the master tells the slave that a data transfer is complete.

Table 4-2 UNIBUS Signal Descriptions (Cont)

Name	Mnemonic	Number of Lines	Function
Data Transfer Signals (Cont)			
Slave sync	SSYN	1	The bus slave asserts and negates SSYN, which is received by the master. When SSYN is asserted during a data transfer, the slave tells the master that it has received the data. The host (slave) asserts SSYN during an interrupt to tell the master that it has received the interrupt vector.
Parity	PA, PB	2	The bus slave asserts the parity error indicators and the master receives them. PA = Ø, PB = Ø: no parity error exists. PA = Ø, PB = 1: parity error exists. The RUX5Ø does not use PA.
Interrupt	INTR	1	The RUX5Ø asserts INTR to the host processor after the RUX5Ø becomes bus master. It signals that an interrupt is to be performed and that an interrupt vector is on the D lines.
Priority Arbitration Signals			
Bus request	BR7:4	4	The RUX5Ø asserts the appropriate BR line to request a bus interrupt.
Bus grant	BG7:4	4	The host processor asserts the appropriate BG line in response to a BR and grants use of bus.
Selection acknowledged	SACK	1	The RUX5Ø asserts SACK in response to a bus grant.

Table 4-2 UNIBUS Signal Descriptions (Cont)

Name	Mnemonic	Number of Lines	Function
Priority Arbitration Signals (Cont)			
Nonprocessor request	NPR	1	The RUX5Ø asserts NPR anytime it is ready to start a data transfer.
Nonprocessor grant	NPG	1	The host processor asserts NPG to the RUX5Ø and grants use of bus for data transfers.
Bus busy	BBSY	1	The current bus master asserts BBSY to all other devices, indicating that the data section is in use.
Initialization Signals			
Initialize	INIT	1	The host processor asserts INIT and starts a subsystem reset operation.
AC low	AC LO	1	AC LO is asserted by any device on the UNIBUS and is received by the RUX5Ø. AC LO indicates possible failure and aborts a RUX5Ø data transfer if in operation.
DC low	DC LO	1	DC LO is asserted by any device on the UNIBUS and is received by the RUX5Ø. DC LO indicates possible failure and causes a subsystem reset operation.

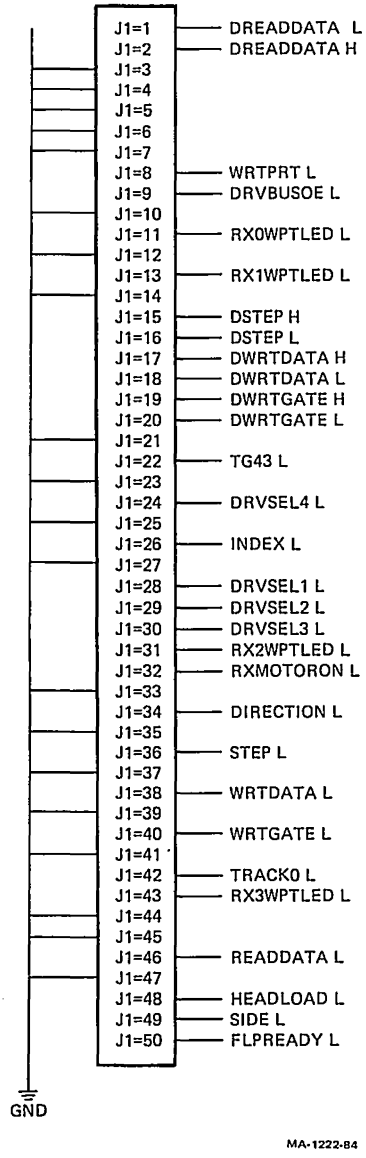


Figure 4-1 Connector J1 Pin Functions

Table 4-3 Drive Interface Signals

Name and State	Pin	Function
DREADDATA L/H	1, 2	Read Data is the data signal input from the drive for long-line configurations. It is a twisted pair signal. Data is a low-going pulse from a minimum of 750 ns to a maximum of 1250 ns for each flux transition.
WRTPRT L	8	Write Protect is sampled whenever a write command is received. When asserted, it terminates the write command and reports a write protected error.
DRVBUSOE L	9	Drive Present indicates that a diskette drive is present and powered up.
RXØWPTLED L	11	Logical unit Ø write protect LED enable
RX1WPTLED L	13	Logical unit 1 write protect LED enable
DSTEP H/L	15, 16	Step Pulse is a twisted pair signal for long-line configurations. It carries a 2-us pulse, which is output to the drive. For every step pulse (6 ms per step) issued, the drive moves one track in a direction determined by the direction control signal (DIRECTION L/H).
DWRTDATA H/L	17, 18	Write Data is a twisted pair signal for long-line configurations. It carries a 250-ns pulse that is generated for each flux transition. Precompensation is generated according to the position of the read/write head. (Precompensation values and track separation will be determined.)
DWRTGATE H/L	19, 20	Write Gate is a twisted pair signal for long-line configurations. This output is made valid before writing can be done on the diskette.
TG43 L	22	Track Greater than 43 indicates that the read/write head is between track 44 and 79, which lowers the write current of the head on the drive during a write operation. The output is valid only during read and write commands.

Table 4-3 Drive Interface Signals (Cont)

Name and State	Pin	Function
DRVSEL4 L	24	Select Unit 4
INDEX L	26	Index Pulse informs the RUX5Ø when the index hole is found on the diskette. Minimum pulse width is 2Ø us.
DRVSEL1 L	28	Select Unit 1
DRVSEL2 L	29	Select Unit 2
DRVSEL3 L	3Ø	Select Unit 3
RX2WPTLED L	31	Logical unit 2 write protect LED enable
RXMOTORON L	32	Motor Power On turns on the drive's spindle motor. The spindle motor takes up to 25 ms to accelerate.
DIRECTION L	34	Stepping Direction Control is active low when stepping in and high when stepping out. It is valid 12 us before the first stepping pulse is generated.
STEP L	36	Step Pulse is a 2 us pulse that is output to the drive. For every step pulse (6 ms per step) issued, the drive moves one track in a direction determined by the direction control signal (DIRECTION L/H).
WRTDATA L	38	Write Data is a 25Ø-ns pulse that is generated for each flux transition. Precompensation is generated according to the position of the read/write head. (Precompensation values and track separation will be determined.)
WRTGATE L	4Ø	Write Gate indicates when the write data is valid. This signal must be active before writing can be done on the diskette.
TRACKØ L	42	Track ØØ informs the RUX5Ø that the read/write head is over track ØØ.
RX3WPTLED L	43	Logical unit 3 write protect LED enable

Table 4-3 Drive Interface Signals (Cont)

Name and State	Pin	Function
READDATA L	46	Read Data is the data signal input from the drive. This signal is a low-going pulse from a minimum of 75Ø ns to a maximum of 125Ø ns for each flux transition.
HEADLOAD L	48	Head Load selects head Ø when high and head 1 when low for double-sided drives.
SIDE L	49	Side Select selects side A when high and side B when low for double-sided drives.
FLPREADY L	5Ø	Floppy Drive Ready indicates that the diskette drive is ready. This signal is sampled before read or write commands. If FLPREADY is low, the read or write operation is done. If FLPREADY is high, the read or write operation is not done and the RUX5Ø aborts the command. When the RUX5Ø is not doing any operation, it checks the ready lines sequentially every second for the status of the drive to indicate volume changes.

CHAPTER 5
TROUBLESHOOTING

5.1 INTRODUCTION

The RUX5Ø has several methods for detecting errors.

- Self-test at power-up
- Self-test at idle states
- Self-test at the beginning of IP register writes
- Diagnostics

5.2 SELF-TEST

The power-up self-test is a set of firmware routines in the RUX5Ø. It verifies that the T-11, the ROM and RAM, and the diskette drive controller are functioning correctly.

The self-test does not test the logic associated with UNIBUS or the drive data transfers. The test selects the drives to indicate their presence. This is the only test for the drives.

Four LEDs on the RUX5Ø give an immediate indication of its pass or fail status, and they pinpoint the test sequence when a failure occurs.

The self-test takes less than 5 seconds to complete. It runs continuously after power-up and during idle states when the RUX5Ø is not serving a drive. During a IP write, the RUX5Ø stops running the self-test and tests the drive control status register (CSR). This test asserts all the drive select LEDs before starting port initialization to show that the RUX5Ø is about to be initialized.

The self-test code is made up of two test routines: T11tst and CSRtst. T11tst tests the T-11, diskette drive controller, and the ROM and RAM. CSRtst tests the drive CSR and asserts all the drives' select and write protect LEDs.

If the LEDs never flash on the drives, check the LEDs on the RUX5Ø to see which test failed. If the self-test does not detect any errors, the RUX5Ø loops on T11tst. CSRtst does not repeat.

CSRtst also runs at the beginning of each IP write or port initialization to show that the RUX5Ø is undergoing port initialization. T11tst does not run at port initialization. Table 5-1 shows the power-up and idle state sequence that the RUX5Ø executes during the self-test.

Table 5-1 Self-Test Sequence

Sequence	LED States	Routine	Description
1	Ø111	CSRtst	Starts at power-up or bus initialization and when a IP write is detected. Tests the drive CSR on the RUX5Ø and flashes the drive select and write protect LEDs on the drives.
2	ØØØ1	T11tst	Starts testing the T-11 commands and address modes.
3	ØØ1Ø	T11tst	Tests the diskette drive controller.
4	ØØ11	T11tst	Tests the interrupt logic diagnostic interrupt.
5	Ø1ØØ	T11tst	Does a checksum on ROM.
6	Ø1Ø1	T11tst	Does a moving inversion test on RAM.
7	ØØØØ	T11tst	Leaves the T11tst routine.
8	Ø***	T11tst	Repeats steps 2 through 7, looping on the T11tst. The three lower LEDs flash in the sequence described above. The flashing may not be very noticeable. Looping in the T11tst stops when the host processor requests port initialization.
9	1ØØ1	--	Part 1 of port initialization is complete.
1Ø	1Ø1Ø	--	Part 2 of port initialization is complete.
11	11ØØ	--	Part 3 of port initialization is complete.
12	1ØØØ	--	Part 4 of port initialization is complete or running MSCP or DUP.

On detecting a fatal error in the operation of the diskette drive controller, T-11, or the ROM and RAM, the self-test loads an error code into the RUX50's SA register. The self-test then loops on a branch instruction to halt the operation of the RUX50. The host processor must reinitialize the RUX50 to start the self-test and port initialization sequence again. Table 5-2 provides the error code definitions that the MSCP receives.

Table 5-2 Generic Error Code Definitions

Error Value Bits 10--0 Bit 15 Set	SA Error Message
1	Envelope/packet read error (parity or timeout)
2	Envelope/packet write error (parity or timeout)
3	RUX50 ROM and RAM parity error
4	RUX50 RAM parity error
5	RUX50 ROM parity error
6	Ring read error (parity or timeout)
7	Ring write error (parity or timeout)
8	Interrupt master failure
9	Host access timeout (higher level protocol dependent)
10	Credit limit exceeded
11	UNIBUS master error
12	RUX50 fatal error
13	Instruction loop timeout
14	Illegal virtual circuit ID
15	Interrupt vector illegal
16	Maintenance read/write invalid region identifier
17	Maintenance write load to nonloadable RUX50
18	RUX50 RAM error (nonparity)
19	Initialization sequence error

Table 5-2 Generic Error Code Definitions (Cont)

Error Value Bits 10--0 Bit 15 Set	SA Error Message
20	Higher level protocol incompatibility error
21	Purge/poll hardware failure
22	Mapping register read failure (parity or timeout)
23	Attempt to set port data transfer mapping when option not present
501	T-11/CPU failure
502	Nonparity ROM checksum error

5.3 DIAGNOSTICS

The diagnostic (ZRQAD0) for the RUX50 consists of two parts, the initialization test and the performance exerciser. The operator cannot select which of these two parts to run. Both parts execute sequentially.

NOTE
For building system exercisers with DECX, the RUX50 module is XRQAD0.

5.3.1 Initialization Test

This test verifies the hardware configuration (as specified by the operator) and brings each drive unit on-line. The initialization test precedes the execution of any other test.

The initialization test runs in the following general sequence.

1. Verifies the presence of each drive unit.
2. Checks the BR level specified by the operator.
3. Initializes each RUX50 configured for testing.
4. Brings each drive unit on-line.
5. Issues one or two MSCP reads to the innermost logical block number (LBN) of each selected diskette to ensure that each drive can seek and be read.

When this test finds fatal or hard drive errors, it drops the appropriate unit. If basic error messages are enabled, then the test prints out the reason for dropping the unit. Failed units are not tested further unless the operator intervenes.

5.3.1.1 Initialization Drop Unit Summary -- During the initialization test, the test drops any drives from the sequence if it cannot be brought on-line or if the operator-specified drive does not match the hardware.

5.3.2 Performance Exerciser

This test exercises the drives by simulating a standard operating system environment. This test shows the operating performance of the drives. It uses random diskette addresses, random word counts, and data patterns. Protected diskettes are subject to read-only operations. Unprotected diskettes can be read or written, depending on the answers the operator gives to the software parameter questions. The test declares an end-of-pass when the specified number of bytes have transferred to all the diskettes.

If a read/write error occurs during this test, then the RUX5Ø initiates an appropriate number of retries. If all retries fail, the test reports a hard error to the host processor, displays an error message on the console terminal, and tallies the error for the summary report. The test drops the unit if the hard error count exceeds the specified limit.

5.3.2.1 Performance Exerciser Drop Unit Summary -- During the exercise, the test drops a unit for one of three reasons.

1. The test finds an incomplete transfer of N megabytes, where N is specified by the operator during software questioning.

2. The test finds that the number of hard errors exceeds the operator-specified limit.

3. The test finds a fatal error.

Units dropped for hard errors can be added to the test cycle later. However, the diagnostic clears the statistics for the added unit to Ø. If the operator specified a transfer limit, in which case the unit was dropped for a soft error, the diagnostic may or may not clear the statistics. This depends on the operator's answer to question 1Ø in Paragraph 5.3.3.2..

5.3.3 Operating Instructions

The operating instructions for this diagnostic are in Chapter 5 of XXDP+ Operator's Manual. The following paragraphs give detailed information on the hardware and software questions that the diagnostic asks before execution.

5.3.3.1 Hardware Configuration Questions -- The following questions collect the parameters necessary to identify each RUX50 subsystem. The diagnostic asks these questions in response to a START command.

1. CHANGE HW (L) ?

   Answer NO to use the prebuilt answers for all hardware questions. This diagnostic tests four units with the default answers listed at the end of the following questions. The answers can be changed anytime with the set-up utility. Answer YES if you want the diagnostic to ask you all the hardware questions.

   NOTE

   The first time you run this diagnostic, answer YES to question 1 to set parameters for the RUX50 subsystem. Then, when you run the diagnostic again without changing the parameters by answering NO to question 1, the diagnostic uses the predefined parameters.

2. NUMBER OF UNITS (D) ?

   This question has no default answer. Answer with the number of units to be tested. You can specify one to four units. This answer determines how many times the diagnostic asks the following questions.

3. IP address (O) 172150 ?

   Enter the address of the RUX50's IP register as addressed by the host processor with memory management turned off. The diagnostic expects an even 16-bit address from 160000 to 177774. The default answer is 172150.

4. Vector (O) 154 ?

   Answer with the interrupt vector of the RUX5Ø described in question 3. You can specify a vector address from 4 to 774. The default answer is 154.

5. BR LEVEL [USUALLY 4-RQDXl 5-RUX5Ø] (O) 4 ?

   Answer with the bus request interrupt level used by the RUX5Ø, which is 5. Levels 4 through 7 are acceptable. The default answer is 4, but the RUX5Ø is jumpered for 5.

6. Drive number (D) Ø ?

   Enter the logical unit number (LUN) for one drive associated with the IP register address in question 3. LUNs range from Ø through 15. The number entered here must be within the range implied by the jumper (LUN Ø--7) on the RUX5Ø. The default answer is Ø.

   NOTE

   Questions 7 and 8 are actually software parameter questions. Because different drives can exist on the subsystem, these questions become unit-by-unit, hardware questions. If selected LBNs are greater than the maximum for a drive, the diagnostic assigns the maximum bounds for that drive.

7. Also run DUP exerciser (L) Y ?

   This question does not apply to the RUX5Ø subsystem. Press **RETURN** to continue to the next question.

8. Test entire customer area of this disk (L) Y ?

   The default answer to this question is YES because the operator usually wants to test the entire diskette media. NO prompts the diagnostic to ask these four questions.

   a. Lower octal word of beginning LBN address (O) Ø ?

      Enter the lower octal word for the starting logical block number (LBN) of the data area you want to test. LBNs range from Ø to 799. The default answer is Ø.

   b. Higher octal word of beginning LBN address (O) Ø ?

      Enter the higher octal word for the starting LBN of the data area you want to test. LBNs range from Ø to 799. The default answer is Ø.

c. Lower octal word of ending LBN address (D) 177777 ?

Answer this question with the lower octal word for the LBN of the last data area you want to test. LBNs range from Ø to 799. The default answer (177777) is maximum (799).

d. Higher octal word of ending LBN address (O) Ø ?

Answer this question with the higher octal word for the LBN of the last data area you want to test.

9. WRITE ON CUSTOMER DATA AREA ON THIS DISK UNIT (L) ?

Answering YES destroys any data on the diskette. The following warning message appears with a confirmation prompt.

** WARNING - CUSTOMER DATA AREA WILL BE OVERWRITTEN! .. CONFIRM (L) ?

The default answer is NO if you decide to bypass the hardware questions.

5.3.3.2 Software Parameter Questions -- The diagnostic asks the following questions in response to the START, RESTART, and CONTINUE commands.

1. CHANGE SW (L) Y ?

Answer NO to bypass the following questions, and answer NO when you first run the diagnostic. NO causes the diagnostic to select the default parameters listed with the following questions. Then, depending on the errors detected, change this answer to YES to alter the parameters and isolate the problem.

2. Enter time as HHMM (Example: 13Ø5 - No leading zeros allowed) (D) Ø ?

Enter the time of day (in 24-hour format). Do not enter numeric values beginning with Ø.

3. Hard Error limit (D) 32 ?

Enter the number of hard errors allowed before the diagnostic drops a unit from testing. A number from 1 to 65535 is acceptable.

4. Transfer limit in megabytes (Ø for "Quick pass") (D) Ø ?

   When the specified number of bytes have transferred to or from a unit, the diagnostic drops the unit from testing. When all units are dropped, the diagnostic indicates an end-of-pass. This method determines how long the diagnostic runs.

   NOTE

   The only other way the diagnostic declares an end-of-pass is if it drops all units because the errors on each exceed the limit. You can abort the diagnostic anytime by holding down the CTRL key and pressing C.

5. Percentage of "Fixed Disk" operations out of total operations (D) 99 ?

   The diagnostic adjusts the numbers associated with usage. It does this according to drive type and percentage to simulate a normal operating environment.

6. Clear statistical tables after printing (L) N ?

   Answering YES causes the diagnostic to clear the statistical fields to Ø after the summary report prints (at end-of-pass or at operator request). Answering NO maintains cumulative totals.

7. Want to rewrite blocks when "Forced Error" detected on reads (L) Y ?

   This question does not apply to the RUX5Ø subsystem. Press **RETURN** to continue to the next question.

8. Do you want to halt on Bad-block Hard Errors (#s 35, 38) (L) Y ?

   This question does not apply to the RUX5Ø subsystem. Press **RETURN** to continue to the next question.

9. Do you want to halt on other Hard Errors (#s 31-34, 36-37, 39-45) (L) Y ?

   When the diagnostic runs with the halt on error (HOE) switch set (START/FLAGS:HOE), the diagnostic halts on any error. Answer NO to this question if you want testing to continue when hard errors are found, even with the HOE switch set.

10. Do you want to halt on Soft Errors (#s 50-54) (L) N ?

When the diagnostic runs with the HOE switch set (START/FLAGS:HOE), the diagnostic halts on any error. Answer NO to this question if you want testing to continue when soft errors are found, even with the HOE switch set.

11. Count each retry on a Read/Write error as a separate Soft error (L) N ?

When the diagnostic finds a read or write error, the RUX50 retries the operation a number of times. If the operation is eventually successful, the diagnostic reports a soft error. The errorlog packet contains the number of retries performed before the operation was successful. Normally, the diagnostic classifies the whole sequence of retries as one soft error. Answer YES to this question if you want the diagnostic to count each retry as a separate soft error.

12. Random seek mode (L) Y ?

Answer YES to make the diagnostic select LBNs randomly. Answer NO to make the diagnostic select LBNs sequentially.

13. Read-compares performed at the Controller (L) Y ?

Answering YES causes all read commands to include the compare modifier. This forces the RUX50 to perform two read operations on the same diskette address and compare the results.

NOTE

The following message appears after you answer question 13.

The remaining questions only apply to unprotected disks

14. Write-compares performed at the Controller (L) N ?

Answering YES changes all write I/O requests to write-compares. After each write, the RUX50 reads the data and compares it to data from the host processor.

15. Check all Writes at Host by reading (L) Y ?

    The diagnostic asks this question if you answered NO to question 14. Answering YES to question 15 causes the host processor to check all writes by reading the data immediately after a write. This process consumes extra computing time and doubles the amount of storage required for writes. Answer YES when you suspect drive write-compares.

16. User-defined data pattern (L) N ?

    Answering YES allows you to define your own data pattern to be used in all write operations. NO allows you to select a predefined data pattern in question 18.

17. Select pre-defined data pattern (Ø for Sequential selection) (D) Ø ?

    Twenty-one predefined data patterns are available. Select one by answering with any number from 1 to 21. An answer of Ø causes the diagnostic to select patterns 1 to 21 sequentially for each write.

18. Number of words in data pattern (D) 16 ? pattern values (O) ?

    The diagnostic asks this question if you answered YES to question 16. The actual bit patterns are entered in octal form.

5.3.4 Diagnostic Error Reporting

The diagnostic recognizes four types of errors: fatal system errors, fatal drive errors, hard errors, and soft errors. Fatal system errors indicate that the diagnostic detected an error in loading or controlling the diagnostic process. Fatal drive errors are fatal to the drive, but testing continues. Hard errors result from soft error retries, an unrecoverable error, or a soft error when retries are not set. Soft errors are media related. The RUX5Ø retries all soft errors. The diagnostic retrieves soft errors from the RUX5Ø via the errorlog capabilities of MSCP.

All general error messages include the type of error (fatal system, fatal drive, hard, or soft) and a unit number. If the error applies to a RUX5Ø, then the message gives only the first unit number of the RUX5Ø.

Basic error messages provide more details about the error. The diagnostic prints all basic error messages along with the diskette address, if applicable. In some cases where a fatal drive error applies to a RUX50, the message prints the RUX50's IP register address.

Extended error messages print the relevant fields of command and end message packets, status codes, SA register contents, and errorlog messages. All values are in octal form.

5.3.4.1 Initialization Errors -- The initialization test reports two types of errors: fatal system errors and fatal drive errors. A fatal system error means that too many units are specified. A fatal system error causes the diagnostic to abort. Fatal drive errors affect only the unit(s) involved. Testing continues on all other units. Fatal drive errors include, but are not limited to, the following failures.

Register existence test failure (no drive present)
Vector test failure
BR level test failure
Initialization sequence failure
On-line failure
Access failure

5.3.4.2 Performance Exerciser Errors -- Most errors reported during this diagnostic originate from MSCP end message packets. The diagnostic translates the status code field to text and prints it as part of the basic error message. Any subcode value follows if extended error messages are enabled.

5.3.4.3 Errorlog Messages -- The contents of the errorlog message from the RUX50 prints as the host processor receives it. You can decipher the message with the MSCP specifications.

5.3.4.4 MSCP Errors -- An MSCP error occurs when the host processor receives an invalid command end message from the RUX50. In this case, the host processor prints the erroneous command with the reason for the error. If extended printouts are enabled, then the host processor prints the entire contents of the end message in octal form without data interpretation.

5.3.4.5 Sample Error Message -- The errors listed by the diagnostic are usually very descriptive and self-explanatory. The following example is a sample error message. It is the extended error message.

```
DISK xxx                                !Disk unit number
Invalid Command                         !Major status code received back
SUB-CODE xxxx                           !Sub-code of given command
COMMAND: xxxx                           !Command given to drive
LBN: xxxxx                              !Logical block number given
BYTE COUNT IN COMMAND xxxxx             !Number of bytes wanted to read
ACTUAL # OF BYTES TRANSFERRED xxxxx     !Number of bytes actually read
```

The status code in an end message has two parts. The first five bits represent the major status code given by the invalid command message. The 11 remaining bits represent the subcode, which indicates the error in the RUX5Ø. The LBN is the logical block number on the diskette that the RUX5Ø was trying to read. The byte count refers to the number of bytes the RUX5Ø was going to read from the LBN. The actual number of bytes transferred refers to the number of bytes read before the error.

5.3.4.6 Performance and Progress Reports -- A summary report prints after each pass of the diagnostic or on demand by the operator. Depending on how you answer software question 6 (Paragraph 5.3.3.2), the diagnostic may or may not clear the statistical fields to Ø after the report prints. Any units added to the test cycle begin with cleared statistics.

Errors listed in this report group into two categories, hard and soft. Each of these categories divides into four more categories, depending on the most probable classification for the error.

- Diskette errors
- Seek (or format) errors
- RUX5Ø or drive errors
- Host processor errors

The following example is a sample summary report. All numeric values are in decimal radix.

```
UNIT     # OF  BYTS  # OF   BYTES    == HRD ERS ==    == SFT ERS ==
 #   TYPE READS READ WRITES WRITTEN DAT SEK DRV HST DAT SEK DRV HST
=== ==== ===== ==== ====== ======= ============== ==============

X   XXX  XXXX  XXXX XXXXX  XXXXXX  X   X   X   X   X   X   X   X
X   XXX  XXXX  XXXX XXXXX  XXXXXX  X   X   X   X   X   X   X   X
X   XXX  XXXX  XXXX XXXXX  XXXXXX  X   X   X   X   X   X   X   X
```

5.3.4.7 Error Codes -- Table 5-3 defines the diagnostic's error codes.

Table 5-3 Error Codes Generated by ZRQA Exerciser

Code	Problem	Fault
Fatal System Error		
1	More than four units are specified.	--
Fatal Drive Errors		
1Ø	Host processor couldn't address RUX5Ø.	Wrong IP address selected at the address given.
11	RUX5Ø didn't interrupt at the interrupt vector given.	Wrong vector address selected.
12	RUX5Ø didn't interrupt at the BR level given.	Wrong BR level selected.
13	Initialization sequence failed.	Either one of the four initialization steps did not receive the correct response from the RUX5Ø,or one of the steps timed out.
14	Fatal error exists in RUX5Ø.	The error bit (bit 15) in the SA register was set.
15	Host processor failed to bring unit on-line.	On-line response had an error code. See also codes 22 and 23.
16	Write protect conflict exists.	The unit was hardware write protected and write operations were requested on the unit.
17	Access to the inner or the outer track failed.	Innermost or outermost track's header may be corrupt.
18	Unit went off-line.	--
19	Drive type is not known.	The version of the exerciser being run does not support this drive type.
2Ø	RUX5Ø failed to send the set controller characteristics command.	Either the unit is off-line or the diagnostic is corrupt because of problems with RAM.

Table 5-3 Error Codes Generated by ZRQA Exerciser (Cont)

Code	Problem	Fault
Fatal Drive Errors (Cont)		
21	RUX5Ø returned wrong end code for the set controller characteristics command.	A problem is in the RUX5Ø microcode or the port/DMA interface.
22	Host processor failed to send on-line command.	Either the unit is off-line or the diagnostic is corrupt because of problems with RAM.
23	RUX5Ø returned wrong end code for the on-line command.	A problem is in the RUX5Ø microcode or the port/DMA interface.
24	Drive went to the available state.	--
Hard Errors		
31	RUX5Ø received an invalid command.	The diagnostic is corrupt because of problems with RAM. A problem is in the RUX5Ø microcode (RAM or ROM) or in the port/DMA interface.
32	Command was aborted by RUX5Ø.	Command timed out in the RUX5Ø.
35	Media format error exists.	--
36	Drive is write protected.	--
37	RUX5Ø read or write compare error exists.	--
38	Data error exists.	CRC error in the data field of a diskette block.
39	Host buffer access error exists.	--
4Ø	Controller error exists.	This problem is difficult to categorize without looking at the error subcode or associated errorlog message.

Table 5-3 Error Codes Generated by ZRQA Exerciser (Cont)

Code	Problem	Fault
Hard Errors (Cont)		
41	Drive error exists.	See code 40.
42	Host write compare error exists.	Error detected when host processor compared the data written and read. May be a problem with the host processor or RUX50 RAM.
43	Message from internal diagnostics exists.	See code 40.
44	RUX50 detected duplicate unit number.	--
45	Unknown end code received.	A problem is in the RUX50 microcode or the port/DMA interface.
Soft Errors		
50	Controller error exists.	See errorlog packet for details. The exact cause may not be evident.
51	Host memory access error exists.	See code 50.
52	Drive transfer error exists.	See code 50.
53	Standard drive interconnect error exists.	See code 50.
54	Small disk error exists.	See code 50.

CHAPTER 6
PROGRAMMING

6.1 INTRODUCTION

The RUX5Ø is a direct memory access (DMA) interface with interrupts and 18-bit extended memory addressing. The interface implements MSCP, therefore, it supports SA and IP registers and host handshaking as specified in the UDA5Ø Programmer's Documentation Kit (QP-9Ø5-GZ).

This chapter discusses the control protocols and bus protocols that the RUX5Ø supports and how it conforms to the protocols. This chapter also defines the programmable and hardware registers.

6.2 CONTROL PROTOCOLS

The RUX5Ø supports two control protocols: mass storage control protocol (MSCP) and the diagnostics and utilities protocol (DUP).

6.2.1 Mass Storage Control Protocol

MSCP is the message-oriented set of rules by which the RUX5Ø communicates with the host processor. MSCP allows the host processor to send a command message and the RUX5Ø to send a response message. The host processor designates an area of memory as a communication area and tells the RUX5Ø the location of this area. The size of the communication area varies and is determined by the host software. For more information on MSCP, see the UDA5Ø Programmer's Documentation Kit (QP-9Ø5-GZ).

Because drives that use 5-1/4-inch diskettes do not have the capabilities of larger storage devices, the RUX5Ø conforms to MSCP, except for the following commands and subcommands.

- The RUX5Ø does not generate the following MSCP message formats.

  Access path attention message (not multiported)
  Duplicate unit number attention message (no unit plug)

- The RUX50 rejects the REPLACE command and all host writes to the replacement and caching table (RCT).

- The RUX50 reports all pertinent device geometry information except for the following information.

  Replacement block numbers (RBNs) per track
  Diagnostic block numbers (DBNs) per unit
  RCT copies
  RCT size

- Because the RUX50 is not cached or multiported, it supports the following functions as "no operations" (NOPs).

  Compare controller data
  Determine access paths
  Flush

6.2.2 Diagnostics and Utilities Protocol

DUP allows communication between the host processor and the diagnostic and utilities server task (DUST). This protocol allows the host processor to request the DUST to load a diagnostic or utility and execute it. During execution, the host processor can ask the DUST about the progress of the diagnostic or utility being executed. The host processor can also abort the diagnostic or utility after unexpected results. For more information on DUP, see the UDA50 Programmer's Documentation Kit (QP-905-GZ).

6.3 BUS PROTOCOLS

Because of the type of devices it controls, the RUX50 supports the following UNIBUS/Q-Bus software services protocol (UQSSP) functions.

- 18-bit bus addressing

- Software-set interrupt vector

- Enhanced diagnostic support (data wraparound (SA) and purge/poll tests)

- Last failure packet

The RUX50 does not support the following UQSSP functions.

- Burst NPR transfer size
- Maintenance read
- Maintenance write

6.4 PROGRAMMABLE REGISTERS

The programmable registers on the RUX50 are the initialize and poll (IP) register and the status and address (SA) register. These registers can be addressed as any memory location.

6.4.1 IP Register

The IP register has a standard UNIBUS address of 772150. The device address jumpers (W1 through W12) on the RUX50 determine this address. (See Paragraph 2.3 for jumper selection information.)

The host processor begins the initialization sequence by issuing a bus initialization signal or by using the IP initialize operation. The IP register is not an actual register. It is simply a circuit that checks for a write at the IP register address. Any write to that address starts the initialization sequence.

6.4.2 SA Register

The SA register has a standard UNIBUS address of 772152. The device address jumpers on the RUX50 determine this address.

The SA register consists of two registers, the SA read register and the SA write register. They are named according to their function with the host processor.

The T-11 writes to the SA read register, while the host processor reads from it. This register passes initialization status to the host processor.

The host processor writes to the SA write register, while the T-11 reads from it. During the initialization sequence, this register passes the MSCP command buffer address and interrupt vector to the T-11.

6.5 INTERNAL HARDWARE REGISTERS

The RUX50 has 24 internal registers to control its hardware functions. The T-11 on the RUX50 accesses these registers. The starting T-11 address for this set of registers is 177400. The registers are write-only and read-only registers. See Table 6-1 for the definitions of these registers.

6.5.1 Load Interrupt Vector

Writing to this register defines the interrupt vector that the RUX50 uses for the host processor.

Table 6-1 Hardware Register Definitions

Address	Mnemonic	Function	Type
177400	LDVECT	Load interrupt vector	Write-only
177402	LDSYSDISP	Load system display	Write-only
177404	LDUBSPTR	Load UNIBUS pointer	Write-only
177406	LDUBSEXT	Load UNIBUS extension	Write-only
177410	CLRERR	Clear DMA error flags	Write-only
177412	WRCSR	Load drive CSR	Write-only
177414	LDXMITBUF	Load transmitter buffer	Write-only
177416	LDDATCSR	Load data CSR	Write-only
177420	--	Reserved	--
177422	--	Reserved	--
177424	--	Reserved	--
177426	--	Reserved	--
177430	WR1793	Diskette drive controller write command	Write-only
177432	WR1793	Diskette drive controller write track	Write-only
177434	WR1793	Diskette drive controller write sector	Write-only
177436	WR1793	Diskette drive controller write data	Write-only
177440	DIAGINTR	Diagnostic interrupt	Write-only
177442	INTRUBS	Interrupt UNIBUS	Write-only
177444	RDSA	Read PIO status/address	Read-only
177446	RDRCVRBUF	Read receive buffer	Read-only
177450	ENLUN	Read LUN	Read-only
177452	RDDEV	Read device type	Read-only

Table 6-1 Hardware Register Definitions (Cont)

Address	Mnemonic	Function	Type
177454	RDCSR	Read drive CSR	Read-only
177456	RDDATCSR	Read data CSR	Read-only
177460	RD1793	Diskette drive controller read data	Read-only
177462	RD1793	Diskette drive controller read sector	Read-only
177464	RD1793	Diskette drive controller read track	Read-only
177466	RD1793	Diskette drive controller read status	Read-only

6.5.2 Load System Display

Writing to this register controls the diagnostic LEDs on the RUX50. Bits 08 through 11 control the LEDs. Writing to all other bits does not affect the RUX50.

Bit	LED
08	1
09	2
10	3
11	4

6.5.3 Load UNIBUS Pointer

This is a 16-bit register that receives the UNIBUS DMA address.

6.5.4 Load UNIBUS Extension

This register receives the two UNIBUS extended address bits: 16 and 17. Writing to bit 00 is for the extended address bit of 16 and bit 01 is for 17. Writing to bits 02 through 15 has no effect.

6.5.5 Clear DMA Error Flags

This is a strobe-only register that clears the DMA error flags. The data has no effect.

6.5.6 Load Drive CSR

Bit	Function
ØØ	Control write protect LED Ø (H turns LED on.)
Ø1	Control write protect LED 1 (H turns LED on.)
Ø2	Control write protect LED 2 (H turns LED on.)
Ø3	Control write protect LED 3 (H turns LED on.)
Ø4	Enable double density (H enables; L disables.)
Ø5	Select precompensation Ø (H)
Ø6	Select precompensation 1 (H)
Ø7	Logical drive select 1 (H selects drive 1.)
Ø8	Logical drive select 2 (H selects drive 2.)
Ø9	Logical drive select 3 (H selects drive 3.)
1Ø	Logical drive select 4 (H selects drive 4.)
11	Motor on (H turns on the drive motor.)
12	Side select (H selects side Ø; L selects side 1.)
13	Load head on selected drive (H loads the read/write head.)
14	Not used
15	Not used

6.5.7 Load Transmitter Buffer

This register holds a 16-bit data word for transfer to the host processor or SA read register.

6.5.8 Load Data CSR

The diskette drive controller (WD1793) master control (bit 9), when written to as a Ø, resets the diskette drive controller chip. The GOLDSA (bit 5), when written to as a 1, initializes the DMA control circuits to perform a DMA transfer to the PIO SA read register from the data buffer. The data buffer must be loaded after a GOLDSA for the SA to be loaded. The GODMAFUBS (bit Ø), when written to as a 1, initializes the DMA control circuits to perform a DMA transfer from the host processor to the data buffer.

Bit	Function
ØØ	GODMAFUBS
Ø5	GOLDSA
Ø9	Diskette drive controller master control

6.5.9 Diskette Drive Controller Write Command

Bits ØØ through Ø7 pass a command to the drive controller on the RUX5Ø. Writing to bits Ø8 through 15 has no effect.

6.5.10 Diskette Drive Controller Write Track

Bits ØØ through Ø7 pass a destination track number to the drive controller on the RUX5Ø. Writing to bits Ø8 through 15 has no effect.

6.5.11 Diskette Drive Controller Write Sector

Bits ØØ through Ø7 pass a destination sector number to the drive controller on the RUX5Ø. Writing to bits Ø8 through 15 has no effect.

6.5.12 Diskette Drive Controller Write Data

Bits ØØ through Ø7 pass data to be written to the diskette by the drive controller on the RUX5Ø. Writing to bits Ø8 through 15 has no effect.

6.5.13 Diagnostic Interrupt

This is a strobe-only register that enables the diagnostic interrupt flag. Writing has no effect.

6.5.14 Interrupt UNIBUS

This is a strobe-only register that enables the UNIBUS interrupt flag. Writing has no effect.

6.5.15 Read PIO Status/Address

This register holds a 16-bit, MSCP-defined word containing status or address data.

6.5.16 Read Receive Buffer

This register holds a 16-bit data word from the data buffer receiver.

6.5.17 Read LUN

This register holds an 8-bit word containing the logical unit number, bits ØØ through Ø7. Jumpers W36 through W29 control these bits. Bits Ø8 through 15 are not used.

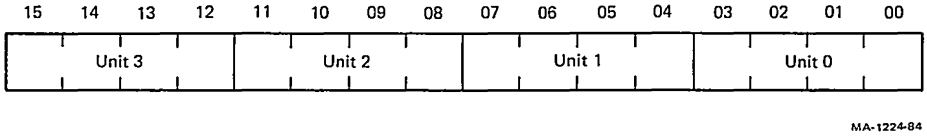


Figure 6-1 Device Type Register

6.5.18 Read Device Type

This register holds a 16-bit word separated into four parts (Figure 6-1). Each part defines the drive type connected for each logical unit.

The RUX5Ø recognizes five device codes that enter this register from jumpers W13 through W28.

Code	Device	Description
ØØØØ	None	No device at this logical unit location
ØØØ1	RX5Ø	Single-sided, double-density, dual unit (96 TPI)
ØØ1Ø	RX18Ø	Single-sided, double-density, dual unit (48 TPI)
ØØ11	Reserved	Double-sided, double-density, single unit (96 TPI)
Ø1ØØ	Reserved	Double-sided, double-density, single unit (48 TPI)

6.5.19 Read Drive CSR

This register contains the status information at the time of the last executed command.

Bit	Function
ØØ	Write protect LED Ø (H = LED Ø on)
Ø1	Write protect LED 1 (H = LED 1 on)
Ø2	Write protect LED 2 (H = LED 2 on)
Ø3	Write protect LED 3 (H = LED 3 on)
Ø4	Double density (H = enabled; L = disabled)
Ø5	Select precompensation Ø (H)
Ø6	Select precompensation 1 (H)
Ø7	Logical drive select 1 (H = drive 1 selected)
Ø8	Logical drive select 2 (H = drive 2 selected)
Ø9	Logical drive select 3 (H = drive 3 selected)
1Ø	Logical drive select 4 (H = drive 4 selected)
11	Motor on (H = drive motor on)
12	Side select (H = side Ø selected; L = side 1 selected)
13	Load head on selected drive (H = read/write head loaded)
14	Selected drive write protected (L)
15	Diskette drive ready (H)

6.5.20 Read Data CSR

This status register provides the following status information to the T-11.

Bit	Function
Ø1	Transmitter full (1 = data transmitter buffer full)
Ø2	Receiver full (1 = data receiver buffer full)
Ø3	Timeout (1 = UNIBUS timeout occurred)
Ø4	Parity error (1 = parity error occurred during DMA operation)
Ø6	Reset DMA control circuits (1 = DMA control circuits hung)
Ø7	DMA done (Ø = DMA not in process)
Ø8	Diskette drive controller master reset status (Ø = reset)

6.5.21 Diskette Drive Controller Read Status

Bits ØØ through Ø7 contain the drive controller's status word. Bits Ø8 through 15 are not used.

6.5.22 Diskette Drive Controller Read Track

Bits ØØ through Ø7 contain the current track number of the read/write head location. Bits Ø8 through 15 are not used.

6.5.23 Diskette Drive Controller Read Sector

Bits ØØ through Ø7 contain the current sector number of the read/write head location. Bits Ø8 through 15 are not used.

6.5.24 Diskette Drive Controller Read Data

Bits ØØ through Ø7 contain the next data word retrieved from the diskette. Bits Ø8 through 15 are not used.

CHAPTER 7
SPECIFICATIONS

7.1 INTRODUCTION

This chapter provides the hardware specifications for the RUX5Ø. It includes information on drives and diskette format to prevent drive compatibility problems.

7.2 PHYSICAL SPECIFICATIONS

Module	Quad-size M7522
Height	26.56 cm (1Ø.46 in)
Width	1.27 cm (Ø.5 in)
Length	22.7Ø cm (8.94 in)
Weight	1 lb

7.3 ELECTRICAL SPECIFICATIONS

Power requirements	+5 Vdc ±5% at 2.9 A (typical) +15 Vdc ±5% at Ø.Ø7 A (typical)
AC bus loads	2.2
DC bus loads	Ø.5

7.4 ENVIRONMENTAL SPECIFICATIONS

Temperature

Storage	5°C to 66°C (4Ø°F to 15Ø°F)
Operating	5°C to 5Ø°C (41°F to 122°F)

Relative humidity

Storage	1Ø% to 95%, noncondensing
Operating	1Ø% to 95%, noncondensing

Altitude

Storage	9.1 km (30,000 ft) maximum
Operating	2.4 km (8000 ft) maximum

Airflow

Operating up to 50°C	Maximum temperature rise across the RUX50 must not exceed 20°C (68°F) input to output.

7.5 OPERATING CHARACTERISTICS

Base device address	772150 (typical, jumper selectable)
Vector	Software selectable (usually set at 154)
Drives per RUX50	Up to four logical units

7.6 DATA FORMATS

The RUX50 can read and write Digital's standard diskette data formats.

7.6.1 Digital's Data Format

In this format, data stores on the diskette in fixed-length blocks. Each track on the diskette divides into sectors with 512 bytes of data on a track. Each sector has two main fields, an ID field and a data field. The ID field stores information about the location of the data. That is, it stores information about which track, side, and sector where the data resides. The data field contains the 512 bytes of data to be read or written by the RUX50. Figure 7-1 represents the format.

7.6.2 Alternate Data Format

The alternate data format on IBM's 5-1/4-inch diskette is similar to Digital's data format. The alternate data format (Figure 7-2) does have two differences.

1. Every track has 4096 bytes of data.
2. The index mark is in gap 1.

"GAP 1"	GAP 1:	(A?) X 4E
REPEAT ONCE PER SECTOR	SYNCH:	(B?) X 00
	IDAM :	3 X A1
		FE
	TRACK:	[PHYSICAL TRACK NUMBER]
	SIDE :	[PHYSICAL SIDE NUMBER]
	SECTR :	[PHYSICAL SECTOR NUMBER]
	SIZE :	02 [TRANSLATION=>512 BYTES PER SECTOR]
	2 CRC :	2 CRC BYTES
	GAP 2 :	22 X 4E
	SYNCH:	12 X 00
	DAM :	3 X A1
		FB
	DATA :	512 X E5
	2 CRC :	2 CRC BYTES
	GAP 3 :	(C?) X 4E
	GAP 4 :	(D?) X 4E

LEGEND: FOR EXAMPLE, DIGITAL's FORMAT WOULD USE:
(A?) = 47
(B?) = 8
(C?) = 48
(D?)~6250 MINUS (THE SUM OF BYTES UP TO GAP 4).

MA-1221-84

Figure 7-1 Digital's Data Format

"GAP 1"	GAP1A:	80 X 4E
	SYNCH:	12 X 00
	INDEX:	3 X C2
		FC
	GAP1B:	50 X 4E
REPEAT ONCE PER SECTOR	SYNCH:	(B?) X 00
	IDAM :	3 X A1
		FE
	TRACK:	[PHYSICAL TRACK NUMBER]
	SIDE :	[PHYSICAL SIDE NUMBER]
	SECTR:	[PHYSICAL SECTOR NUMBER]
	SIZE :	02 [TRANSLATION=>512 BYTES PER SECTOR]
	2 CRC :	2 CRC BYTES
	GAP 2 :	22 X 4E
	SYNCH :	12 X 00
	DAM :	3 X A1
		FB
	DATA :	512 X E5
	2 CRC :	2 CRC BYTES
	GAP 3 :	(C?) X 4E
	GAP 4 :	(D?) X 4E

LEGEND: FOR EXAMPLE, IBM's FORMATTED DISKETTE WITH 8 TRACKS:
(B?) = 12
(C?) = 64
(D?)~ 6250 MINUS (THE SUM OF BYTES UP TO GAP 4).

MA-1225-84

Figure 7-2 Alternate Data Format

7.7 SUPPORTED DRIVES

The RUX5Ø has a device type register, which can be set through jumpers. During the initialization sequence, the host processor reads the device type register and translates each logical unit with the following characteristics (Figure 7-3).

DEVICE TYPE REGISTER	DEVICE NAME	DENSITY	HEADS PER UNIT	DISKS PER UNIT	TRACKS AND (TPI)	STEP RATE	HEAD SETTLE TIME	TUNNEL ERASE TIME	MOTOR DELAY
0000	NONE	=	=	=	=	=	=	=	=
0001	RX50	DOUBLE	1	2	80 (96)	12 ms	50 ms	3 ms	0.5 s
0010	RX180	DOUBLE	1	2	40 (48)	20 ms	30 ms	3 ms	1 s
0011	FUTURE	DOUBLE	2	1	80 (96)	12 ms	50 ms	3 ms	0.5 s
0100	FUTURE	DOUBLE	2	1	40 (48)	20 ms	50 ms	3 ms	0.5 s

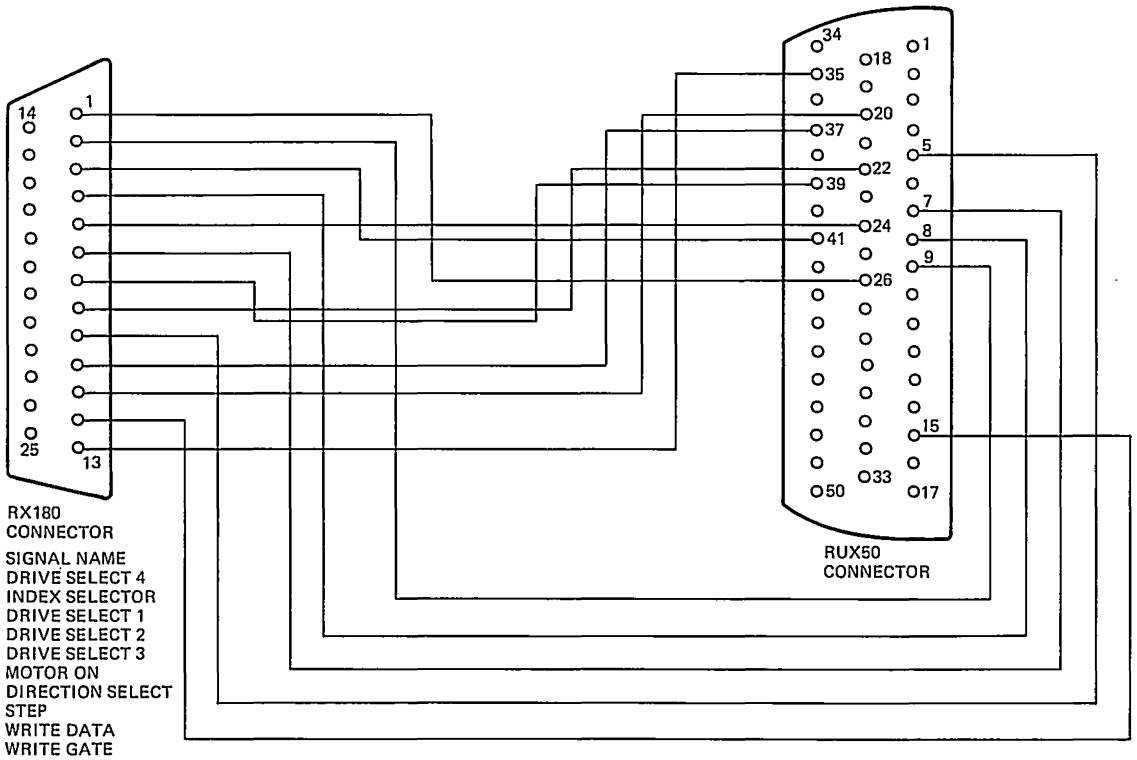
MA-1223-84

Figure 7-3 Supported Drives

APPENDIX
RX18Ø CABLING

The RUX5Ø can interface RX18Ø diskette drives. RX18Øs are single-density, single-sided, dual-diskette, (5-1/4-inch) drives. Digital no longer sells or supports RX18Øs.

To connect a RX18Ø to the RUX5Ø, configure the RUX5Ø as described in Chapter 2 for a single-sided, dual-drive (48 TPI) controller. Build a cable to connect the RX18Ø and RUX5Ø by following the point-to-point connector chart in Figure A-1.



RX180	SIGNAL NAME
1	DRIVE SELECT 4
2	INDEX SELECTOR
3	DRIVE SELECT 1
4	DRIVE SELECT 2
5	DRIVE SELECT 3
6	MOTOR ON
7	DIRECTION SELECT
8	STEP
9	WRITE DATA
10	WRITE GATE
11	TRACK 00
12	WRITE PROTECT
13	READ DATA
14–25	GROUND

Figure A-1 RX18Ø Cabling

Digital Equipment Corporation • Maynard, MA 01754