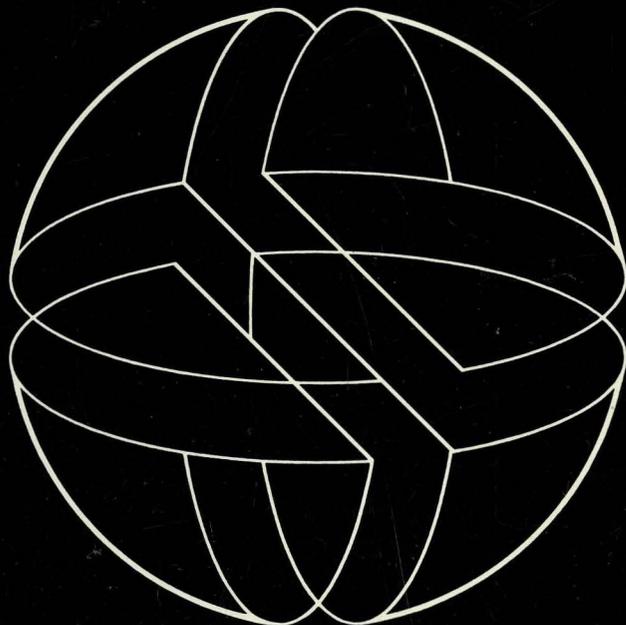


DECmate

Technical
Description



digital

DECmate

Technical Description

Copyright © 1981 by Digital Equipment Corporation
All rights reserved

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.

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

DIGITAL	DECsystem-10	MASSBUS
DEC	DECSYSTEM-20	OMNIBUS
PDP	DIBOL	OS/8
DECUS	EDUSYSTEM	RSTS
UNIBUS	VAX	RSX
	VMS	IAS

CONTENTS

CHAPTER 1 SYSTEM DESCRIPTION

1.1	General.....	1-1
1.2	DECmate Overview	1-1
1.2.1	CPU Board	1-1
1.2.1.1	Microprocessor	1-1
1.2.1.2	Control Panel Memory	1-2
1.2.1.3	Main Memory.....	1-3
1.2.1.4	CRT Control.....	1-3
1.2.1.5	Real Time Clock.....	1-3
1.2.1.6	RX278 Diskette Control.....	1-3
1.2.1.7	Printer Port.....	1-3
1.2.2	Keyboard	1-3
1.2.3	Power Supply	1-3
1.2.4	Video Monitor.....	1-3
1.3	Options.....	1-3
1.3.1	DP278 Communication Option	1-3
1.3.2	RL278 Disk Controller Option	1-4
1.3.3	Extended Language Option.....	1-4

CHAPTER 2 CPU FUNCTIONAL DESCRIPTION

2.1	General.....	2-1
2.2	Microprocessor.....	2-1
2.3	I/O Controllers	2-2
2.4	Main Memory	2-3
2.5	Control Panel Memory.....	2-3
2.6	Terminal I/O Emulation.....	2-6
2.7	Keyboard Control.....	2-7
2.8	Video Display Control.....	2-9

CHAPTER 3 PRINTER PORT

3.1	General.....	3-1
3.2	Functional Block Diagram	3-1
3.2.1	Programmable Baud Rate Generator	3-2
3.2.2	Printer Device Code.....	3-2
3.2.3	Loopback	3-2
3.2.4	Automatic Product Test (APT)	3-3
3.2.4.1	Framing Error.....	3-3
3.2.4.2	APT Connections.....	3-3

CHAPTER 4 FLOPPY DISK INTERFACE

4.1	General.....	4-1
4.2	Data Flow.....	4-3
4.2.1	CPU-To-Disk Transfer.....	4-4
4.2.2	Disk-To-CPU Transfer.....	4-4

CHAPTER 5 DP278 COMMUNICATION OPTION

5.1	General.....	5-1
5.2	Physical Description	5-1
5.2.1	Interrupts	5-1
5.2.1.1	Receiver Interrupts (2).....	5-1
5.2.1.2	Transmitter Interrupts (2).....	5-1
5.2.1.3	Modem Change Interrupt.....	5-1
5.3	Port Description	5-1
5.3.1	Standard Communication Device (SCD).....	5-2
5.3.1.1	Asynchronous Operation	5-3
5.3.1.2	Synchronous Operation	5-3
5.3.1.3	Clocking.....	5-3
5.3.1.4	Maintenance.....	5-3
5.3.1.5	Unused Pins.....	5-3
5.3.2	Extended Synchronous Device (ESD).....	5-3
5.3.2.1	Clocking.....	5-4
5.3.2.2	Maintenance.....	5-4
5.3.3	Port Control Registers	5-4
5.3.4	Port Status Registers	5-4
5.3.5	Modem Control.....	5-4
5.3.6	Line Handling.....	5-4
5.3.7	Modem Interrupt Handling	5-6
5.3.8	Slew Rate.....	5-6
5.3.9	I/O Structure	5-6
5.3.10	Terminal Emulation Hardware.....	5-6
5.3.11	Power Consumption.....	5-7
5.3.12	Switches and Jumpers.....	5-7
5.4	Functional Block Diagram	5-7
5.4.1	BDX Bus.....	5-7
5.4.2	LDX Bus.....	5-9

CHAPTER 6 RL278 DISK CONTROLLER OPTION

6.1	General.....	6-1
6.2	Functional Description.....	6-2
6.2.1	Operation	6-3
6.3	Command Execution.....	6-3
6.3.1	Maintenance	6-3
6.3.2	Reset	6-4
6.3.3	Get Status	6-5
6.3.4	Seek	6-5
6.3.5	Read Header.....	6-5
6.3.6	Write Data	6-8
6.3.7	Read Data.....	6-9
6.3.8	Read Data (Without Header Check).....	6-10

FIGURES

1-1	DECmate Block Diagram	1-2
2-1	Microprocessor, Memory, and I/O Control.....	2-1
2-2	Memory Organization.....	2-4
2-3	Memory Read-Modify-Write.....	2-4
2-4	Control Panel Memory.....	2-5
2-5	Keyboard Control Block Diagram	2-8
2-6	Character Display	2-9
2-7	Video Display Control Block Diagram	2-10
3-1	Printer Port Functional Block Diagram	3-1
4-1	RX278 Functional Block Diagram	4-2
4-2	Interface Timing Diagram	4-3
5-1	DP278 Functional Block Diagram.....	5-10
6-1	RL278 Controller Block Diagram	6-1
6-2	RL278 Controller Functional Block Diagram	6-2
6-3	Maintenance Command.....	6-4
6-4	Get Status Command.....	6-5
6-5	Seek Command	6-6
6-6	Read Header Command	6-7
6-7	Write Data Command.....	6-8
6-8	Read Data Command	6-9
6-9	Read Data (Without Header Check) Command	6-10

TABLES

3-1	Baud Rate Selection.....	3-2
4-1	Function Codes for LCD IOTs	4-1
5-1	Modem Signal Definitions	5-5
5-2	Switch and Jumper Configurations.....	5-8

CHAPTER 1 SYSTEM DESCRIPTION

1.1 GENERAL

This chapter provides an overview of the DECmate system. Later chapters provide more detailed information about the major DECmate components. Consult the *VT278 Maintenance Print Set (MP00900)* for further information.

1.2 DECmate SYSTEM OVERVIEW

Figure 1-1 is a diagram of the DECmate system, which consists of a VT278 video terminal and optional disk subsystem. The following paragraphs describe each major functional block of the diagram.

1.2.1 CPU Board

The VT278 CPU board controls all receive, send, and display functions of the terminal. It receives data from the following components.

- Keyboard
- Diskette drives
- Printer (XON-XOFF data control characters)
- Communication lines (if DP278 option is installed)
- RL02 disk drives (if RL278 option is installed)

The CPU board sends command information and data to the following components.

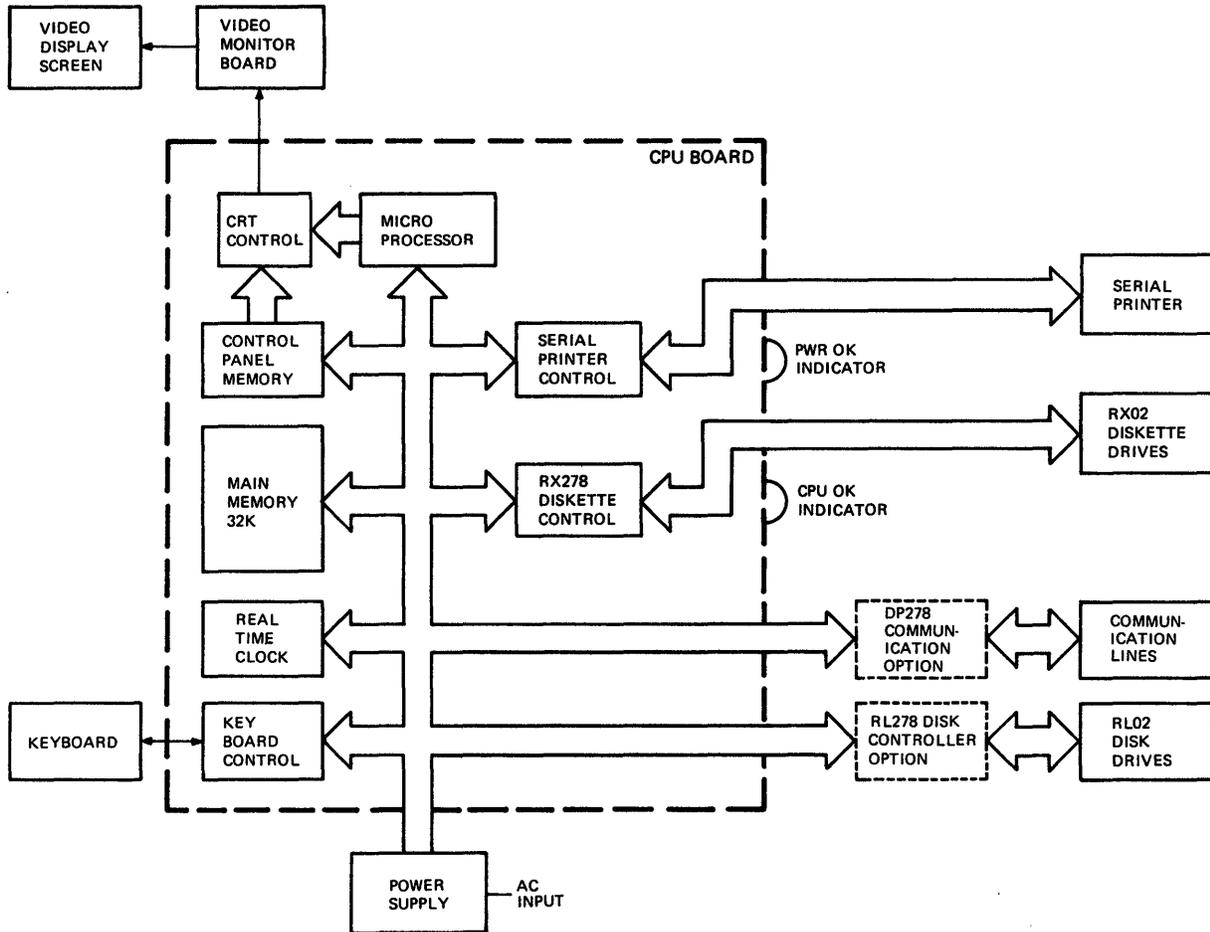
- Keyboard
- Video monitor (for display on CRT screen)
- Serial printer
- Diskette drives
- Communication lines (if DP278 option is installed)
- RL02 disk drives (if RL278 option is installed)

The CPU board has two indicator lights. Both are visible through the rear panel of the VT278.

- **PWR OK** lights to indicate that dc voltages are present and within tolerance.
- **CPU OK** blinks at a rate of 1 Hz to indicate that CPU is functional.

The CPU board contains the following major circuits.

1.2.1.1 Microprocessor – The 6120 microprocessor is a high-speed CMOS processor that executes the VT278 instruction set. (Refer to the *DECmate System Architecture Manual* for a detailed description.)



MA-6773

Figure 1-1 DECmate Block Diagram

The microprocessor has two purposes.

- It executes the user program from the 16K or 32K word main memory.
- It uses a second memory address space (control panel memory) to control video display and keyboard. A special class of interrupts suspends the user program and transfers control to code in control panel memory. This operation is transparent to the user and user program.

1.2.1.2 Control Panel Memory – Control panel memory is a 32K word memory address space in addition to the normal 32K word range of the VT278. This additional memory space allows rapid context switching. The 32K words of control panel memory address space perform the following functions.

- Self test
- Terminal I/O emulation
- Floppy and auxiliary bootstrap
- Optional foreign language display
- Video display buffer memory
- CRT control registers

1.2.1.3 Main Memory – The basic VT278 includes 32K 12-bit words of MOS random access memory.

1.2.1.4 CRT Control – The VT278 display unit consists of a portion of control panel memory, a dedicated CRT controller, and a CRT. It displays either 24 rows of 80 characters, or 14 rows of 132 characters. Increased intensity, underscore, blink, or reverse video can be assigned in any combination to characters on an individual basis. Display functions are controlled by ESCape Sequences passed from the user program as part of the normal terminal I/O. The ESCape Sequences are listed in the *DECmate System Architecture Manual*.

1.2.1.5 Real Time Clock – The real time clock supplies the programmer with an event clock. The clock is crystal-controlled with a frequency of 100 Hz and a basic accuracy of 0.01 percent. The instruction set is given in the *DECmate System Architecture Manual*.

1.2.1.6 RX278 Diskette Control – A single 37-pin connector at the rear of the VT278 allows connection to one or two dual diskette drives. The instruction set is given in the *DECmate System Architecture Manual*.

1.2.1.7 Printer Port – A 25-pin connector at the rear of the VT278 allows a variety of EIA serial printers to be used. The baud rate is set to 9600 when power is first turned on. It may be changed to a speed in the range of 50–19200 baud under program control. The instruction set is in the *DECmate System Architecture Manual*.

1.2.2 Keyboard

The keyboard is connected to the VT278 CPU board by a 3-wire cable. Two wires of the cable supply +12 Vdc and ground to power keyboard logic. The third wire is a bidirectional serial communication line. It carries command information (requests for keyboard data, LED indicator information, and bell commands) from the CPU board to the keyboard. It also carries a unique address for each typed key from the keyboard to the CPU board.

1.2.3 Power Supply

The power supply accepts either 120 Vac or 220/240 Vac input. A slide switch located on the rear panel selects voltage input. A second switch selects either a 50 Hz or 60 Hz input frequency. The power supply generates +5 Vdc and ± 12 Vdc for use by terminal logic.

1.2.4 Video Monitor

The VT278 video monitor accepts data from the CPU board and power from the power supply. From these inputs, it develops the signals necessary to display data on the CRT screen. The video monitor contains all video adjustment controls.

1.3 OPTIONS

The DP278 and RL278 are single-board options. They are installed directly into the VT278 terminal enclosure. The extended language option is a ROM (read only memory) which inserts into the CPU board.

1.3.1 DP278 Communication Option

The DP278 communication option provides two synchronous/asynchronous serial communication lines with full modem control. In asynchronous mode, line speed is fully programmable. In synchronous mode, line speed is set by the modem or VT278 to a maximum of 9600 baud. The DP278 contains firmware to simulate a simple I/O terminal with or without full modem control. The two versions of the DP278 are explained below.

- **DP278-A** provides two programmable communication lines with full modem control.
- **DP278-B** is similar to DP278-A, but has additional circuitry to support bit synchronous (BITSTUFF) operation.

Run-time characteristics for each port may be selected as follows.

Character length – 5, 6, 7, or 8-bit

Number of stop bits – 1, 1.5, or 2

Parity – odd, even, or none

Operating mode – half-duplex or full-duplex

Clocking – synchronous or asynchronous

Protocol – byte-oriented or optional BITSTUFF

Line speed (baud) – 50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600, 19200

The instruction set for the DP278 is given in the *DECmate System Architecture Manual*.

1.3.2 RL278 Disk Controller Option

The RL278 disk controller option controls up to four RL02 disk drives. The RL278 offers fully programmable DMA addressing. Up to 4096 words (8-bit or 12-bit) may be transferred at one time. The instruction set for the RL278 is given in the *DECmate System Architecture Manual*.

1.3.3 Extended Language Option

The extended language option is a ROM added to the basic CPU board. It becomes part of the 32K addressing range of control panel memory and implements specific keyboard and display translation for up to seven languages.

CHAPTER 2 CPU FUNCTIONAL DESCRIPTION

2.1 GENERAL

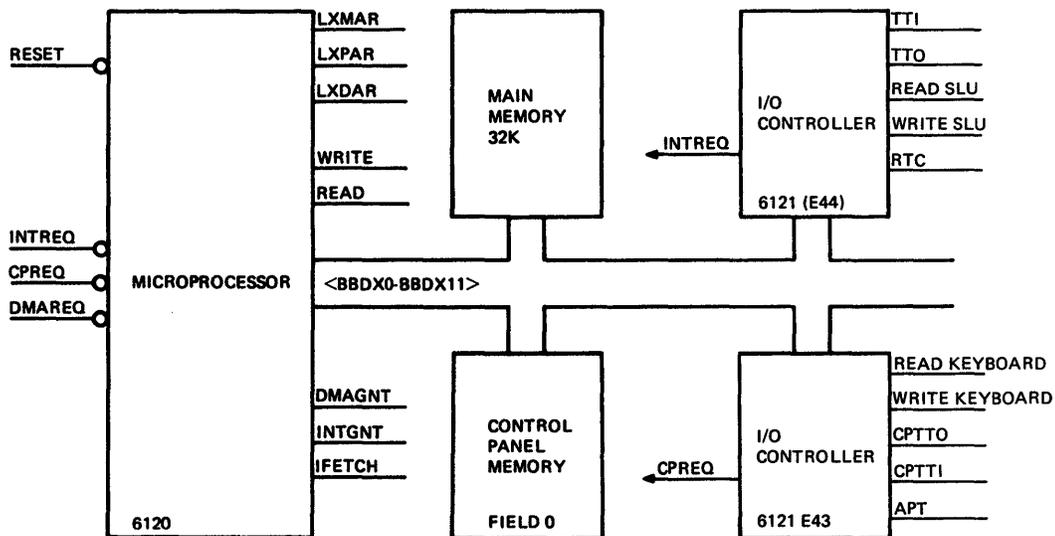
The CPU board contains a microprocessor, memory, and I/O circuits. These circuits control VT278 operation. The following CPU board circuits are discussed in this chapter.

- Microprocessor
- I/O controllers
- Main memory
- Control panel memory
- Terminal I/O emulation
- Keyboard control
- Video display control

2.2 MICROPROCESSOR

The 6120 CMOS microprocessor is a 40-pin, integrated circuit contained on the CPU board. It operates using 12-bit, two's complement arithmetic. The microprocessor controls VT278 operation by executing system and user programs consisting of 12-bit instructions.

The 6120 microprocessor (Figure 2-1) communicates with main memory, control panel memory, and peripheral devices as determined by three signals.



MA-6886

Figure 2-1 Microprocessor, Memory, and I/O Control

LXMAR – main memory
LXPAR – control panel memory
LXDAR – peripheral devices

The signals **READ** and **WRITE**, in conjunction with the above signals, determine the exact operation performed by the microprocessor. For instance, both **LXMAR** and **READ** are asserted during a read main memory operation.

The 6120 microprocessor has three mechanisms for suspending normal program execution.

1. Normal interrupt requests (**INTREQ**) to main memory interrupt handling routines.
2. Control panel memory interrupt requests (**CPREQ**) cause the 6120 to access control panel memory (which contains bootstrap, self test, and other programs necessary to emulate a video terminal).
3. Direct memory access requests (**DMAREQ**), also called data breaks, permit data transfer between high-speed mass storage devices and memory (used only when **RL278** disk option is installed).

The 6120 microprocessor acknowledges main memory and control panel memory interrupts by issuing an interrupt grant signal (**INTGNT**). DMA requests are acknowledged with a direct memory access grant signal (**DMAGNT**). The grant signal, in both cases, may be used to alert the interrupting device that the 6120 is ready to service the interrupt.

The 6120 microprocessor has an internal priority for servicing requests. The state of the priority network determines the next operation. The priority is listed below.

RESET – initializes all internal flags and buffers (6120 remains in reset state as long as **RESET** line is held low)

RUN/HLT – starts and stops microprocessor operation

DMAREQ – requests DMA data transfer (only if **RL278** option is installed)

CPREQ – requests jump into one of the areas of control panel memory

INTREQ – requests jump into main memory interrupt handling routines

IFETCH – if no other requests are pending, fetches the next program instruction as defined by the program counter

2.3 I/O CONTROLLERS

Interrupts to the 6120 microprocessor are largely controlled by two 6121 programmable I/O controllers. The I/O controllers provide basic control and enabling signals for the devices which they control. Each 6121 controls five devices.

I/O controllers permit a pseudo-vector method of recognizing interrupts. The programmer can either poll the devices to verify the interrupt or enable the vector through software. When the vector mode is enabled, the device code of the interrupting device is loaded into the 6120 microprocessor accumulator. Then, it is only necessary for the programmer to read the accumulator to find the interrupting device.

The 6121 I/O controllers are programmed automatically by microcode in control panel memory shortly after power-up. The I/O controller in location E44 is programmed first with the following codes. (Device priority is controlled internally by the 6121 with the first device having the highest priority.)

- Device 03** – TTI buffer
- Device 04** – TTO buffer
- Device 32** – read serial line unit
- Device 33** – write serial line unit
- Device 13** – real time clock

Each time one of these devices needs servicing by the 6120 microprocessor, the 6121 issues an interrupt signal. This I/O controller supplies the normal main memory interrupt. When the 6120 microprocessor receives this interrupt, it jumps to an interrupt handling routine. After servicing the interrupting device, the 6120 returns to the user program.

The I/O controller in location E43 generates the higher level control panel memory interrupts discussed earlier. The control panel 6121 I/O controller is programmed as follows.

- Device 11** – read keyboard
- Device 05** – write keyboard
- Device 06** – control panel memory TTO buffer
- Device 07** – control panel memory TTI buffer
- Device 12** – Automatic Product Test (APT) (used for factory testing)

Each time one of these devices needs servicing by the 6120 microprocessor, the 6121 issues a control panel memory interrupt. The microprocessor leaves the user program, enters a control panel memory routine to service the device, then returns to the user program.

2.4 MAIN MEMORY

Main memory is a storage area for the user program. The VT278 stores 32K 12-bit words. The 6120 microprocessor has an addressing range of 32K (32,768) 12-bit words. Memory is organized into 4K word groups called fields (Figure 2-2). Within each memory field, each location has a unique 4-digit octal address (0000₈-7777₈).

Each field is further divided into 40₈ (32) pages of 200₈ (128) words. The most significant 5 bits of the 12-bit memory address define the page number within a field. The remaining seven bits define the address of a memory location within a page.

Figure 2-3 shows a timing diagram of a main memory read-modify-write cycle. LXMAR determines that the 6120 communicates with main memory.

2.5 CONTROL PANEL MEMORY

Control panel memory provides 32K words of memory address space in addition to the 32K words of main memory. It contains bootstrap loader, self test, and other programs and storage space necessary to emulate a video terminal. Control panel memory supplies memory space for routine functions that normally reside in main memory. Thus, it allows main memory to be used only for the user program.

Control panel memory is organized into fields in the same manner as main memory. Figure 2-4 illustrates how these fields are used.

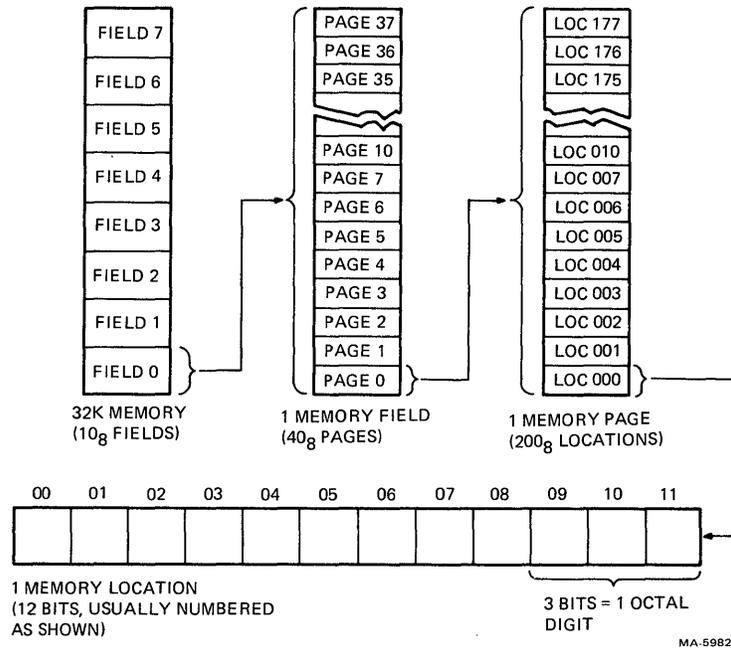


Figure 2-2 Memory Organization

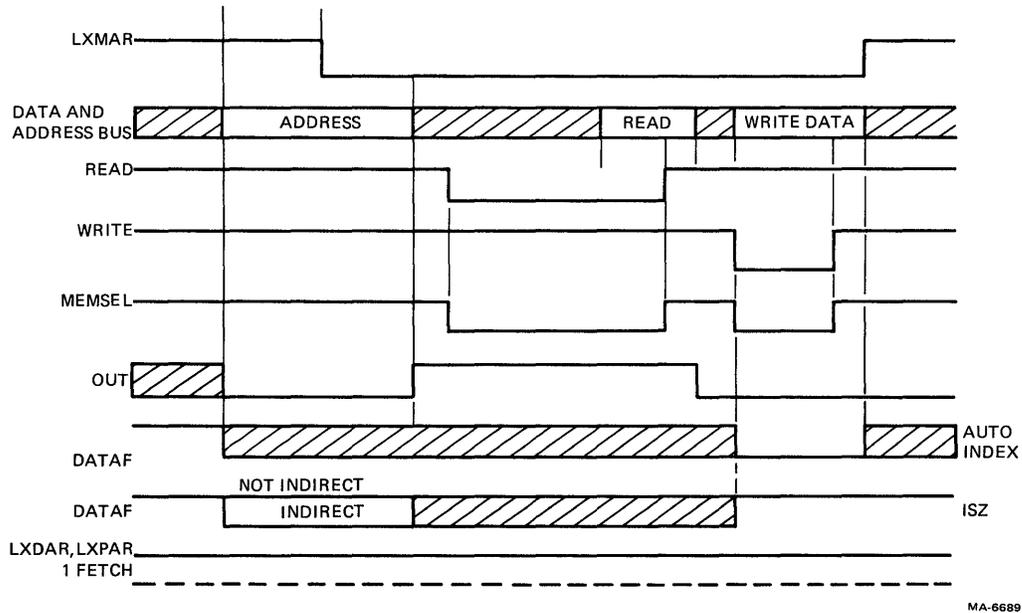
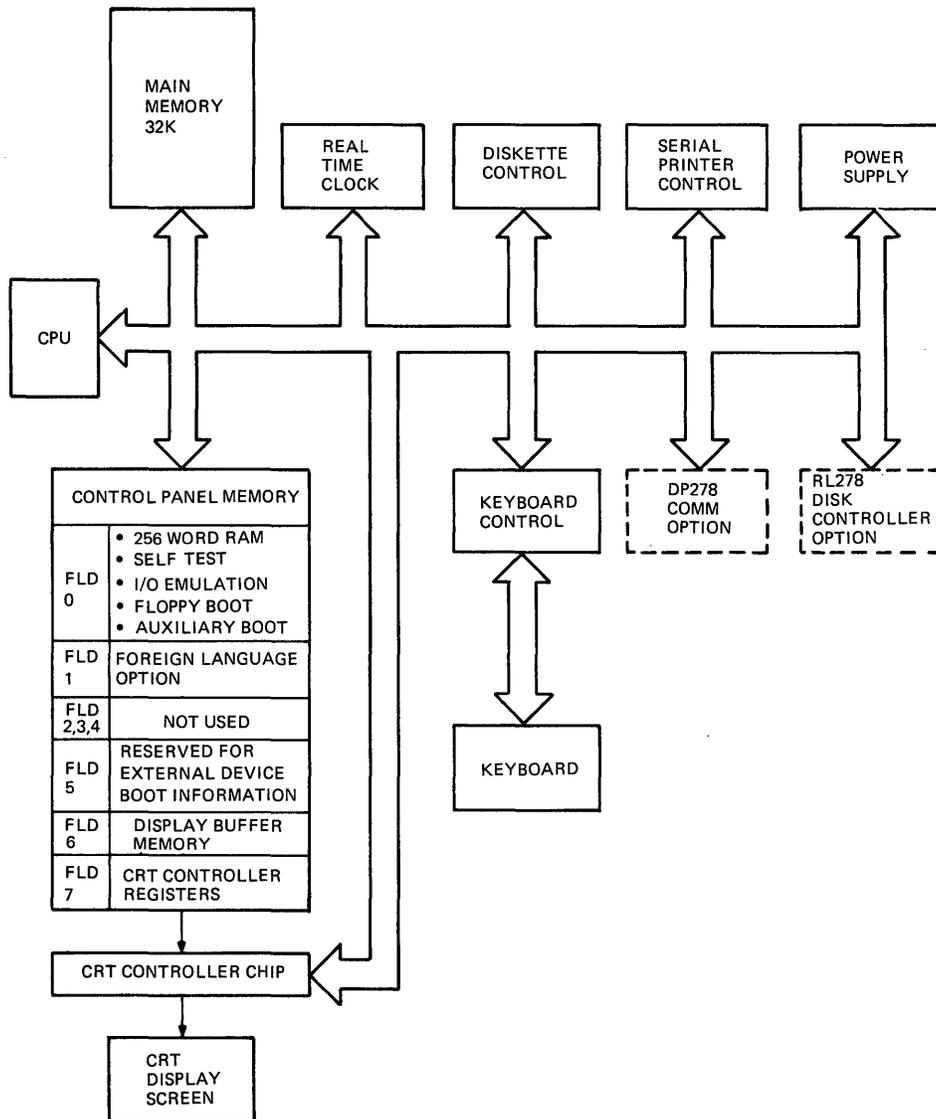


Figure 2-3 Memory Read-Modify-Write



MA-6033

Figure 2-4 Control Panel Memory

Field 0 contains 256 words of random access memory (RAM) and 3K words of read only memory (ROM). The RAM is used for scratchpad memory (TTI and TTO buffers) and for temporary storage of characters waiting to be displayed on the video screen. The ROM contains microcode for the resident self test diagnostics, diskette bootstrap loader program, and terminal emulation.

Field 1 is the extended language option, used only when that option is installed. A ROM containing microcode is added to the CPU board in order to allow operation of natural language keyboards.

Fields 2, 3, and 4 are not used.

Field 5 is reserved for external boot information. The bootstrap code of the DP278 communication and RL278 disk controller options is located in ROMs on the options themselves. When these options are installed, this code is accessed by field 5 of control panel memory.

Field 6 is the display buffer memory. The display buffer is a 2K (2048) by 12-bit RAM. Each memory location relates to a character position on the video screen. Each location stores an 8-bit character code and 4-bits of information for the display attributes of bold, blink, underline, and reverse video.

Field 7 contains nine locations used to program the dedicated CRT controller registers. These registers set up the operating parameters for the CRT screen display.

Control panel memory routines can be entered in four ways.

1. POWER-ON executes self test routines.
2. HALT (processor instruction code 7402) displays halt message and enters SET-UP.
3. PANEL REQUEST COMMAND executes panel request function.

PR0 reads any locations in display buffer memory.

PR1 transfers blocks of data directly to display buffer memory.

PR2 simulates keyboard entry into set-up mode.

PR3 transfers directly to a specified control panel memory location.

4. BOOTSTRAP executes appropriate bootstrap routine.

2.6 TERMINAL I/O EMULATION

The VT278 differs from many other small machines. The console terminal keyboard and display are closely coupled to the 6120 microprocessor through the processor bus by the 6121 I/O controllers. No separate keyboard/display devices communicate to the processor by a serial line. Instead, the VT278 uses control panel memory to emulate such a device.

The microprocessor supports two kinds of interrupts: normal interrupts (hardware subroutine call to location 0000) and control panel interrupts. A control panel interrupt saves the main memory program counter without affecting flags or memory address extension registers. Control panel memory contains ROM code (transparent to the user) which intercepts normal terminal I/O instructions. Thus, any data transfer to the screen or from the keyboard involves a control panel interrupt to the microprocessor.

If a TLS (output a character to the terminal) instruction is issued, the I/O controller generates a control panel interrupt. ROM code in control panel memory transfers the character to a temporary input buffer located in control panel memory. This buffer holds a maximum of 32 characters. The TTO flag sets after the character is stored in the buffer and exits from control panel memory to the user program.

Main Memory

Control Panel Memory

TLS control panel interrupt —————> Identify routine, store character in input buffer, set TTO flag

←————— Return to main memory

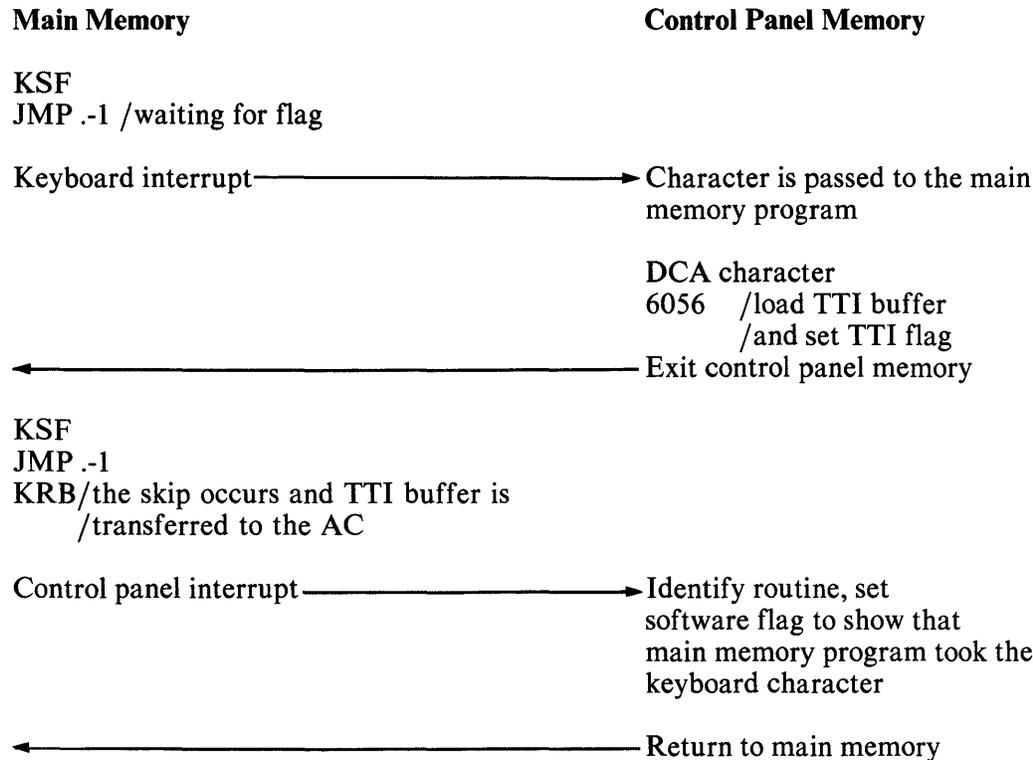
TSF /this instruction will skip immediately

JMP .-1

Field 6 of control panel memory is dedicated to the display buffer. A dedicated CRT controller chip accesses this area in order to provide a conventional 24 row by 80 column (or 14 row by 132 column) display. During the vertical retrace time of the display, characters in the temporary input buffer are transferred by control panel memory code to the display buffer memory.

At the end of each display buffer update a keyboard scan is initiated to locate pressed keys. Keys found during the scan cause an interrupt to control panel memory. After a decoding process, valid characters are loaded one at a time into the control panel memory TTI buffer. A flag sets to interrupt the user program as each character is loaded. The characters are then passed to the user program.

Another control panel interrupt is caused when the user program takes a character. This interrupt indicates to the keyboard handling code that the TTI buffer is empty and ready to accept another character.

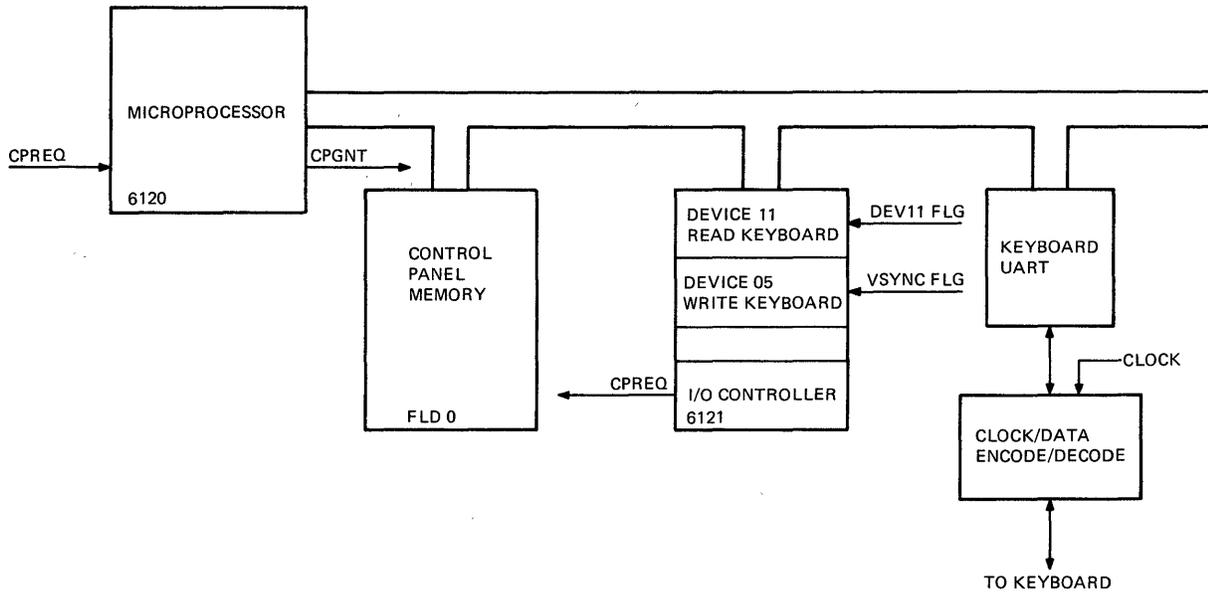


The effective maximum rate of data transfer to the screen during terminal I/O emulation is 19K baud. The display/keyboard service code in control panel memory is designed so that the normal main memory interrupt system is disabled for no more than 1 millisecond following any one control panel interrupt. Detailed programming considerations concerning this 1 millisecond control panel interrupt are explained throughout the programming sections of this manual.

2.7 KEYBOARD CONTROL

The keyboard is connected to the VT278 CPU board by a 3-wire cable. One wire supplies +12 volts to power keyboard logic. One wire supplies ground to the keyboard. One wire is a bidirectional communication line.

A clock signal and data are transferred from the CPU board to the keyboard. The clock signal supplies timing to the keyboard, while the data initiates a keyboard scan and lights the various LED indicators. An address code to identify typed keys is sent from the keyboard to the CPU board.



MA-6687

Figure 2-5 Keyboard Control Block Diagram

Figure 2-5 shows a keyboard control block diagram. Operation occurs as follows.

1. The VSYNC FLG signal initiates a CPREQ to the 6120 microprocessor at the end of each raster scan, every 16.67 milliseconds (20 milliseconds for 50 Hz systems).
2. The 6120 determines by polling that device 05 (write keyboard) interrupted.
3. An initiate keyboard scan character is sent to the keyboard UART.
4. The UART serializes the character and sends it to the keyboard.
5. The keyboard scans the key matrix.
6. Keys found to be pressed send a key address to the UART.
7. The signal DEV 11 FLG (read keyboard) is received by the I/O controller.
8. A CPREQ for device 11 is developed.
9. The 6120 grants the interrupt. The key address is taken from UART to a control panel memory TTI buffer.
10. The TTI flag sets, initiating a main memory interrupt.
11. The 6120 grants the interrupt. It transfers the character from control panel memory to the TTI buffer.

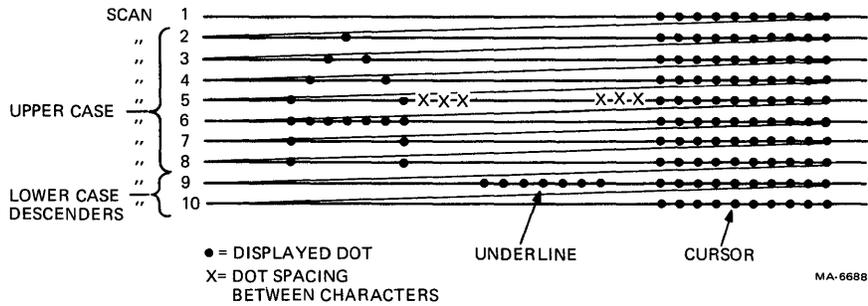


Figure 2-6 Character Display

The major circuit component is the UART (universal asynchronous receiver transmitter). It changes parallel to serial data, and serial to parallel data. This allows data transfer between the parallel 6120 and serial keyboard.

At the end of each vertical scan of the video screen a keyboard scan is initiated. This occurs 60 times per second when the ac input frequency is 60 Hz, and 50 times per second at 50 Hz. The signal VSYNC FLG causes a CPREQ to be generated by the control panel memory 6121. The interrupt sends an initiate keyboard scan character to the UART. The character is serialized and sent to the keyboard. The keyboard receives the character and begins a keyboard scan at the keyboard clock rate.

Any key found to be pressed sends a unique key address through the keyboard cable to the UART. This key address causes the UART to develop the signal DEV 11 FLG. DEV 11 FLG causes the 6121 to generate a control panel memory interrupt. Control panel memory microcode then takes the key address from the UART, translates it to the appropriate character code, and places it into a TTI buffer located in field 0 of control panel memory.

To eliminate the problem of key bounce, a key must be down for two successive scans before it is considered pressed. Control panel memory routines verify this condition before passing the character to the user program.

2.8 VIDEO DISPLAY CONTROL

Characters are displayed on the CRT screen by an electron beam moving quickly across the screen while moving slowly downward (Figure 2-6). The electron beam turns off and on in order to make the dot patterns that form characters.

The entire area scanned by the electron beam is called the raster. The CRT supplies the horizontal and vertical timing signals to control raster scanning. The raster is scanned 60 times per second (50 times per second at 50 Hz). At 60 Hz, the time to complete one raster scan is 16.67 milliseconds.

Initial screen display is set for 24 rows by 80 characters. Scrolling characteristics are forward or backward, jump scroll only. The VT278 provides the following attributes in any combination on a character by character basis.

- Blinking
- Underline
- Bold
- Reverse video

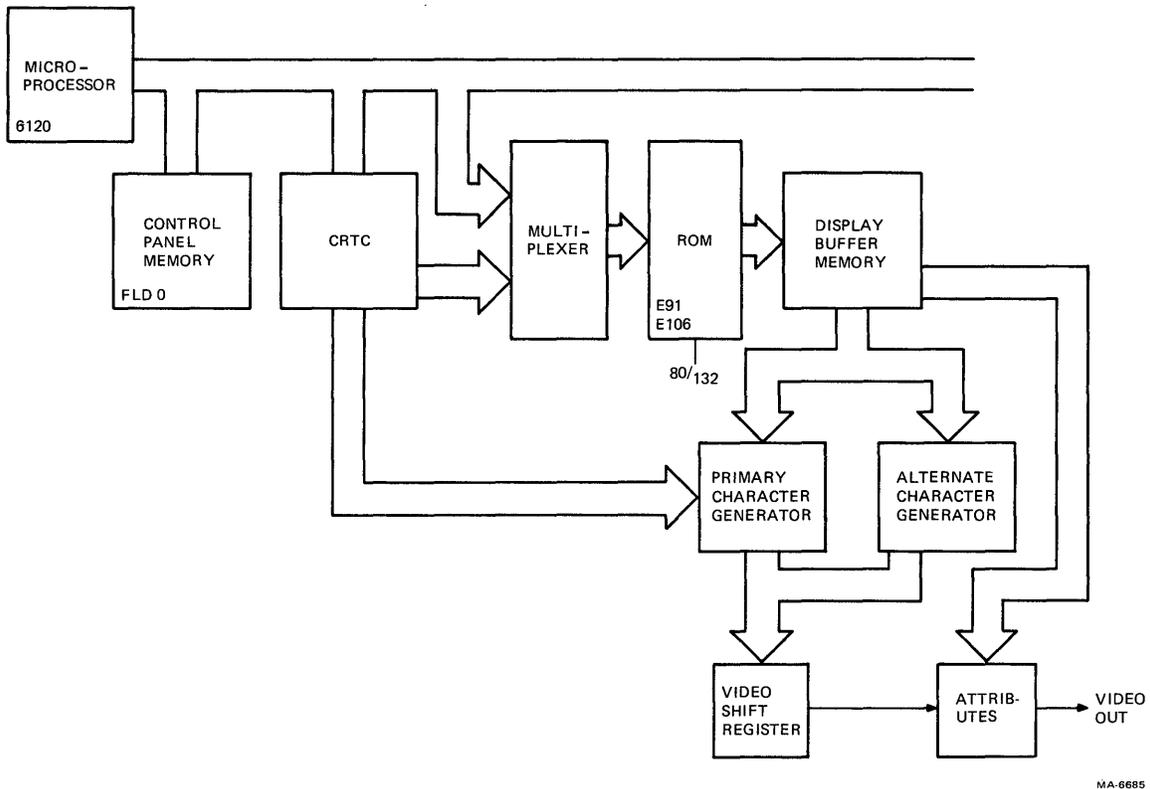


Figure 2-7 Video Display Control Block Diagram

The display format can be changed under program control to 14 rows by 132 characters. Attributes and scroll characteristics remain unchanged.

Figure 2-7 is a block diagram of video display circuitry. The operation of video display circuitry is described below.

1. The CRT controller (CRTC) is programmed at power-up.
2. The CRTC begins raster scan using display buffer memory data (display buffer is empty at this time).
3. The program issues TLS instructions. Characters are placed into the control panel memory TTO buffer (CPTTO).
4. At the end of the next scan, characters are moved into the display buffer memory by the 6120 microprocessor.
5. The CRTC controls the next raster scan using data in the display buffer memory.
6. The dot matrix code for each character is found in the character generator ROM.
7. The dot matrix code is loaded into a shift register.

8. The resulting serial code is combined with any assigned attributes for that character.
9. The video signal is timed with the beam and sent to the CRT for display.

The CRTC controls the video display. It is a programmable controller dedicated to displaying data on the video screen. It is automatically programmed with the display characteristics of the VT278 shortly after power-up. The CRTC programmable registers are in field 7 of control panel memory.

The data to be displayed on the video screen is stored in the display buffer memory. The display buffer memory is 2K 12-bit locations of storage addressed as field 6 of control panel memory. In the normal display mode, 24 rows by 80 characters (1920 locations) are used. In the alternate mode, 14 rows by 132 characters (1848 locations) are used. ROMs E91 and E106 control the 80 or 132 character addressing of display buffer memory.

The 6120 microprocessor or the CRTC can address display buffer memory. This is controlled by the multiplexer (Figure 2-6). The 6120 loads characters to be displayed into display buffer memory, as directed by the user program in main memory. The CRTC addresses display buffer memory in order to display the characters on the video screen at the proper time.

After being programmed, the CRTC steps through the display buffer memory locations, causing the electron beam to scan the raster. At the proper time, each 12-bit character is read from display buffer memory.

Eight of the bits represent the character to be displayed. The dot matrix code for the character is contained in the character generator ROM (either primary or alternate). The dot matrix code is sent to the video shift register which supplies a serial dot matrix video signal.

The remaining 4 bits of the 12-bit character define the character attributes. These four bits are decoded and the attributes assigned on a character by character basis.

Two important functions take place at the end of each raster scan.

1. A keyboard scan is initiated (Paragraph 2.7).
2. Characters located in the control panel memory TTO buffer are moved to the display buffer.

The control panel memory TTO buffer holds a maximum of 32 characters for display. The buffer is emptied at the end of each raster scan, every 16.67 milliseconds (20 milliseconds at 50 Hz). The characters are moved to the display buffer memory when no dots are being formed on the screen. This eliminates screen flicker while changes are being made to the display buffer.

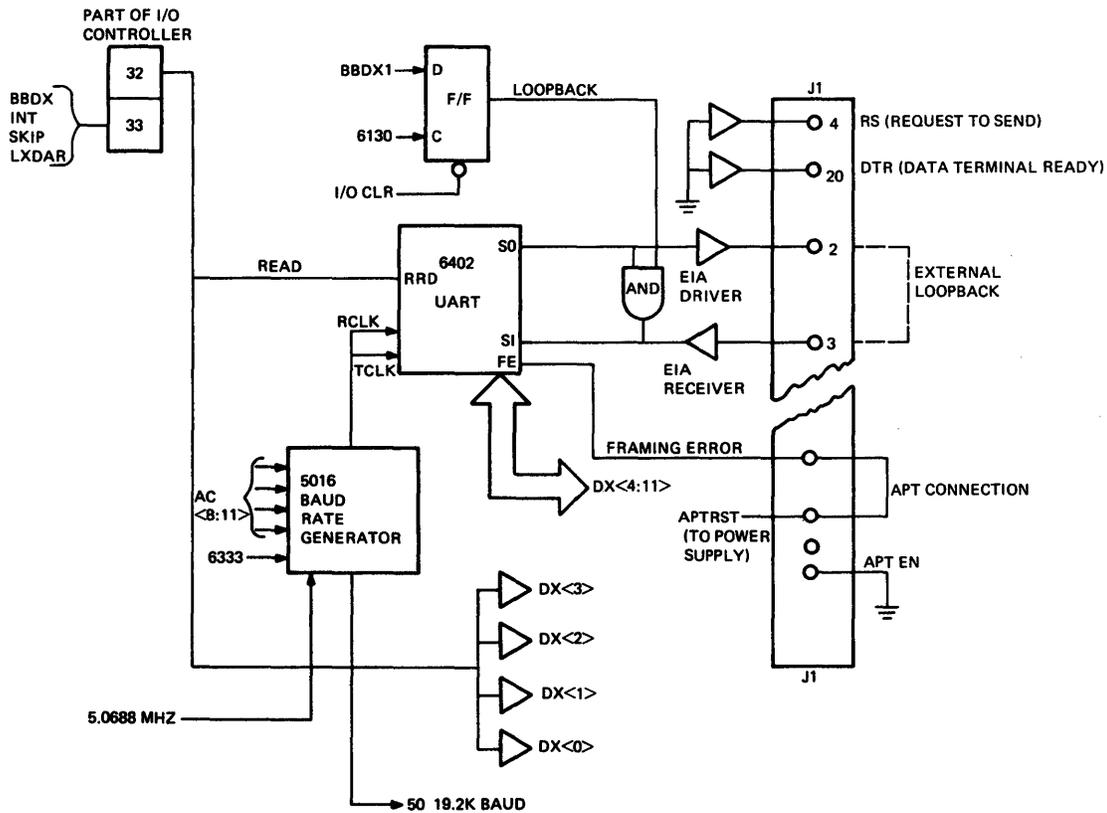
CHAPTER 3 PRINTER PORT

3.1 GENERAL

The printer port allows a serial printer to be connected to the VT278 video terminal. The printer port uses a standard universal asynchronous receiver transmitter (UART), configured for 8 data bits, 1 stop bit, and no parity bit. If the serial printer connected to the printer port is not compatible with this configuration, it must be reconfigured.

3.2 FUNCTIONAL BLOCK DIAGRAM

Figure 3-1 is a functional block diagram of the printer port. The following paragraphs describe its major elements.



MA-7187

Figure 3-1 Printer Port Functional Block Diagram

Table 3-1 Baud Rate Selection

Control Bit State				Baud Rate
8	9	10	11	
0	0	0	0	50
0	0	0	1	75
0	0	1	0	110
0	0	1	1	134.5
0	1	0	0	150
0	1	0	1	300
0	1	1	0	600
0	1	1	1	1200
1	0	0	0	1800
1	0	0	1	2000
1	0	1	0	2400
1	0	1	1	3600
1	1	0	0	4800
1	1	0	1	7200
1	1	1	0	9600
1	1	1	1	19200

3.2.1 Programmable Baud Rate Generator

The only programmable feature of the printer port is the line speed. It can range from 50K to 19.2K baud. Baud rate selection is accomplished by the 6333 IOT instruction which loads 4 bits (AC8-AC11) into the programmable baud rate generator (Figure 3-1). The baud rate generator receives a timing input of 5.0688 MHz. Table 3-1 shows the baud rate selection and state of the accumulator control bits for each baud rate.

3.2.2 Printer Device Code

There are 32 printer port device codes for receiving data, and 33 for transmitting data. There are two 6121 I/O controller chips: E43 and E44. These chips connect to the DX bus and are programmed with I/O information during power-up. The I/O information includes device codes of devices controlled by I/O controller chip E44. If IOT 6326 or 6336 is issued to device 32 or 33 (printer port), E44 recognizes it as an IOT and outputs the appropriate signal. This signal is ANDed with READ or WRITE to control the read or write operation performed by UART. If a receive data IOT (6326) is issued, the contents of the UART receiver register are transferred to the accumulator on the CPU board. If a transmit data IOT (6336) is received, data is transferred from the accumulator to the UART.

3.2.3 Loopback

The printer port allows both internal and external loopback. Internal loopback makes system testing easier by feeding the data output by the UART back into the input without going through the external connector. Internal loopback ties the TTL output of the UART to the TTL input of the UART. This is accomplished by setting the loopback flip-flop. The flip-flop is set by IOT 6130 and the assertion of AC bit 1 (BBDX1).

External loopback connects pins 2 and 3. This connects UART data output to UART data input. However, the EIA driver and EIA receiver are now in the loop. The external loopback thus allows the EIA driver and receiver to be tested.

The UART provides only 8 bits of data while the accumulator on the CPU is 12 bits wide. Consequently, when transferring data (bits DX4–DX11) from the UART to the accumulator, bits DX0 through DX3 are forced to 0s.

3.2.4 Automatic Product Test (APT)

APT provides a factory test function by connecting the VT278 to a remote PDP-11. Dedicated pins on J1 provide connections for APT and APT EN line.

To determine if the system is running under APT, an IOT is issued to test the APT EN line. If the line is grounded, an APT loader routine is initiated.

The APT loader routine eliminates much of the self test, and the VT278 is controlled at the remote processor.

3.2.4.1 Framing Error – When UART detects a character larger than expected, it asserts the FRAMING ERROR signal. Normally, UART expects an 8-bit character followed by a stop bit. The APT connector ties the FRAMING ERROR line to the APRST (automatic product restart) line in the power supply.

A break signal from the host computer is recognized as a framing error by the printer port. The FRAMING ERROR signal is asserted and is applied to the power supply where it internally simulates power supply turn-on and turn-off. The power supply steps through the restart routine and asserts DC OK. This simulated process is interpreted by the VT278 as power turn-on.

3.2.4.2 APT Connections – Three normally unused pins on the printer connector implement automatic testing of the VT278 using the APT system.

Pin 11 – Normally open, grounded by the APT system to indicate that the VT278 is under APT control.

Pin 18 – Connected to the framing error output of the UART.

Pin 25 – Connected to pin 18 by the special APT connector causing FRAMING ERROR signal, produced when APT attempts to gain control of the system under test, to initiate a RESET operation within VT278.

Using the APT interface signals on the printer connector, the APT computer can gain control of the VT278 regardless of the state of the machine, and initiate loading of diagnostics through the printer port.

CHAPTER 4 FLOPPY DISK INTERFACE

4.1 GENERAL

The RX278 floppy disk drive interface controls data transfer between the VT278 and a maximum of two dual floppy disk drive units (RX02). The RX278 can use single density or double density recordings, making it compatible only with Digital Equipment Corporation products.

The floppy disk interface logic decodes instruction bits 3 through 11. This selects a floppy disk drive system for transmitting or receiving data from the VT278 and for controlling all functions to be performed. Bit 11 selects the floppy disk drive unit when an SEL (6750) instruction is performed. Drive pair A is selected when bit 11 is 0. Drive pair B is selected when bit 11 is 1. Bits 8 through 10 select the function to be performed when an LCD (6751) instruction is executed. Bit states and corresponding functions are listed in Table 4-1.

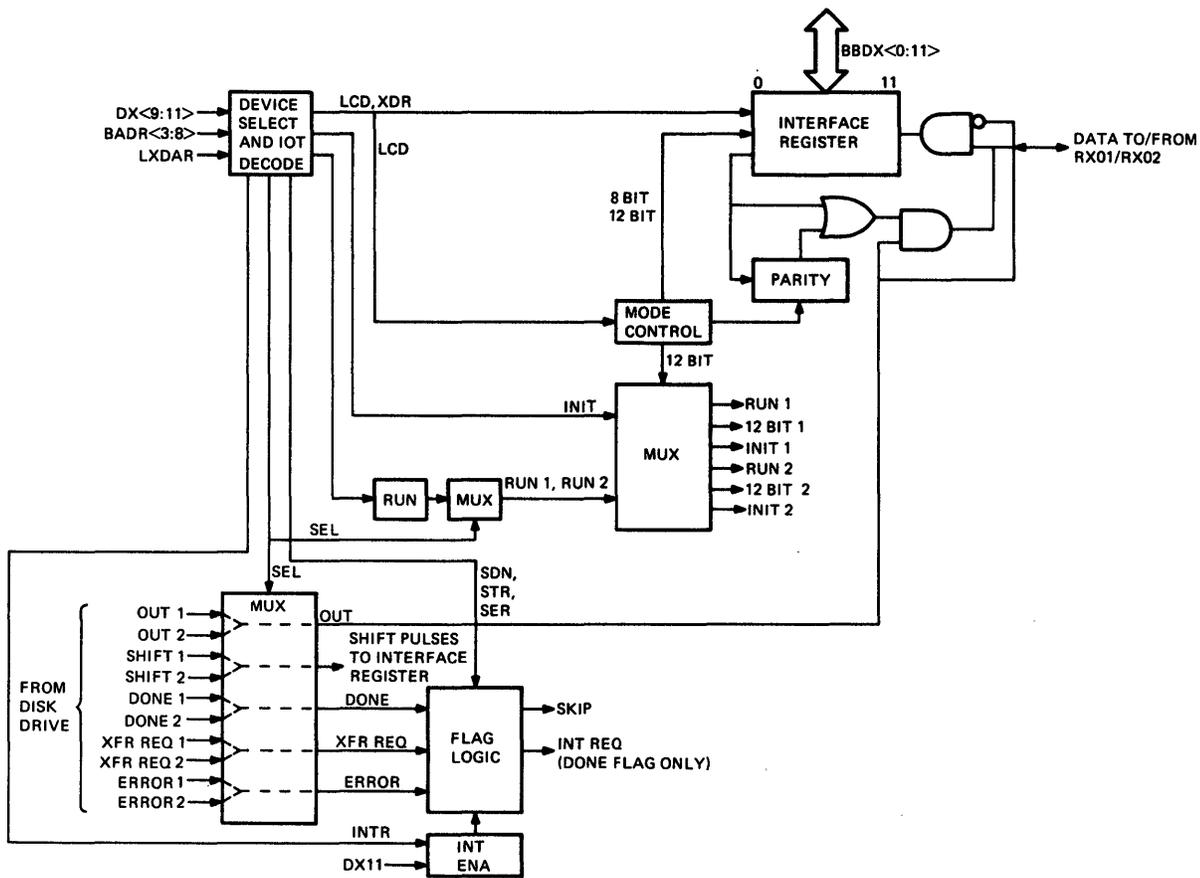
The floppy disk drive interface provides two modes of data transfer: 12-bit and 8-bit. The state of microprocessor accumulator register bit 5 determines which mode is used.

When bit 5 is 0, the 12-bit mode is selected. In single density units, 64 12-bit words can be written in a diskette sector (two sectors equal one page of information). In double density units, 128 12-bit words can be written in a diskette sector (one sector equals one page of information).

When bit 5 is 1, the 8-bit mode is selected. Bits 0 through 3 are forced to 0s. In single density mode, 128 8-bit bytes can be written in each sector. In double density mode, 256 8-bit bytes can be written in each sector. When going from single density to double density, the storage capacity of a diskette is doubled. Bit 7 of the accumulator selects the floppy disk drive unit during an LCD (6751). Drive 0 is selected when bit 7 is 0, and drive 1 is selected when bit 7 is 1.

Table 4-1 Function Codes for LCD IOTs

Bit Code			Function
8	9	10	
0	0	0	Fill buffer
0	0	1	Empty buffer
0	1	0	Write sector
0	1	1	Read sector
1	0	0	Not used
1	0	1	Read status
1	1	0	Write deleted data sector
1	1	1	Read error register



MA-7189

Figure 4-1 RX278 Functional Block Diagram

Figure 4-1 is a block diagram of the RX278 interface circuits. Since it is possible to connect two floppy disks to the interface, the RUN, OUT, SHIFT, XFR REQ, and ERROR signals are duplicated in Figure 4-1. Figure 4-2 shows the interface timing sequence.

The RUN1 and RUN2 signals alert the floppy disk that the VT278 is ready for the next data transfer.

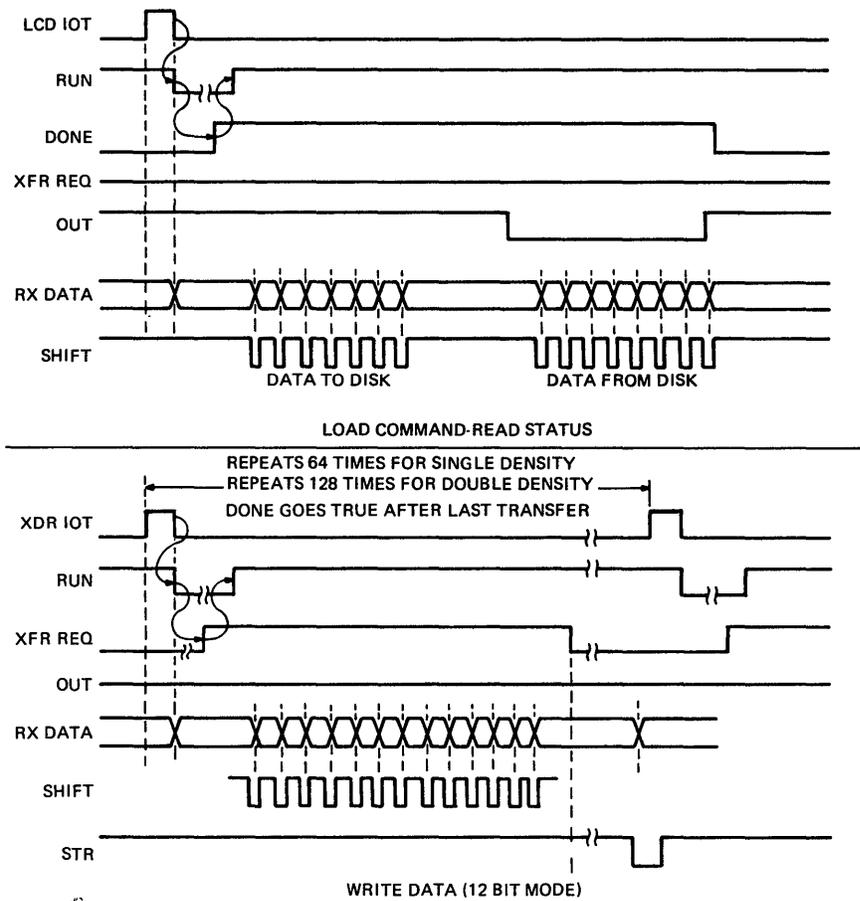
The OUT1 and OUT2 signals determine the direction of the data transfer. If OUT L is asserted, data is transferred from the floppy to the CPU. If OUT H is asserted, data is transferred from the CPU to the floppy.

The SHIFT1 and SHIFT2 signals are generated in the floppy and shift the bits through the interface register.

The XFR REQ1 and XFR REQ2 signals indicate that the floppy is ready to accept a data word (write) or has sent a word to the CPU (read).

The ERROR1 and ERROR2 signals, indicate an error is present. The error sets the DONE flag.

The DONE, XFR REQ, and ERROR signals from the floppy are checked with individual skip IOTs. The SDN (skip on done) IOT checks the DONE flag, the STR (skip on transfer request) IOT checks



MA-0459

Figure 4-2 Interface Timing Diagram

the XFR REQ flag, and the SER (skip on error) IOT checks the ERROR flag. The DONE line is connected to the interrupt bus and, if programmed with INT ENA (interrupt enable), it causes an interrupt to occur. The INT ENA flip-flop sets if an INTR IOT is issued and DX11 is set. If DX11 resets, the INT ENA flip-flop resets. Normal data transfers are handled by the XFR REQ logic and do not cause interrupts.

The RUN flip-flop sets when an IOT is issued, alerting the floppy that it should transmit or receive data. The INIT line forces the floppy to initialize.

4.2 DATA FLOW

Data flow from the CPU to the floppy occurs as follows. The data word contained in the CPU accumulator is parallel-loaded into the interface register by the BBDX bus. The bits are serially shifted out of the interface register, bit 0 first. Shift pulses generated in the floppy perform serial data shifting. Parity is generated and the data is supplied to the floppy. The floppy checks parity but does not generate it.

Data flow from the floppy to the CPU occurs as follows. The serial stream of data is applied to the interface register. The first data bit is applied to bit 11 of the interface register. Bits are serially shifted until the first bit is in bit 0 (12-bit mode) or bit 4 (8-bit mode) of the interface register. The data is then parallel-loaded to the CPU accumulator by the BBDX bus.

4.2.1 CPU-To-Disk Transfer

When data is transferred to the floppy disk drive system from the microprocessor, the floppy disk drive system sets the Transfer Request (XFR REQ) flag, requesting the first data word. The XDR command loads the data word from the microprocessor accumulator into the interface register. This asserts the RUN line, indicating to the floppy disk drive system that a word is available. The floppy disk drive system serially transfers the word to the drive by issuing 8 or 12 shift pulses (depending on the selected mode).

Upon receipt of the entire data word, the floppy disk drive system issues another TR, requesting the next data word. The XDR command loads the data word from the microprocessor accumulator into the interface register. The transfer process is repeated until the sector buffer has been loaded. (In single density operation, 64 data transfers for 12-bit mode, 128 transfers for 8-bit mode. In double density operation, 128 data transfers for 12-bit mode, 256 transfers for 8-bit mode.)

After the sector buffer fills, the floppy disk drive system sets the Done flag, indicating that the function has been completed.

SKIP L asserts if any skip commands (STR, SER, SDN) are decoded by the DEV SEL/IOT decode circuit, and the corresponding flag has been asserted.

4.2.2 Disk-To-CPU Transfer

When data is transmitted from the floppy disk drive system to the microprocessor, the disk drive asserts the out line and issues 8 or 12 shift pulses (depending on the selected mode). The floppy disk drive system issues a XFR REQ flag to indicate that the first word is in the interface register, and that a request has been made for a data transfer from the disk drive to the microprocessor. After the flag has been tested and cleared, the word is transferred to the microprocessor accumulator by the next XDR command. The XDR command causes the RUN line to assert. This indicates to the disk drive that the CPU has accepted the word.

After transferring the next data word into the interface register, the XFR REQ flag sets again, and the transfer process is repeated until the entire contents of the floppy disk drive system buffer register has been transferred. The DONE flag then sets, indicating the end of the transfer function.

As an example of the data flow, assume the VT278 issues a load command (LCD) with an empty buffer function code (001₂). The LCD command causes RUN to assert, and the floppy goes from the done state to the busy state. Since the RUN line is asserted, the floppy interprets the LCD as a command and, outputs a data word to the interface register.

The VT278 issues an XDR command, causing the data in the interface register to transfer to the CPU accumulator through the BBDX lines. This XDR reasserts the RUN line and the process continues until the entire sector is transferred. Each data word requires a XFR REQ to load it into the interface register and an XDR command to transfer it from the interface register to the accumulator. When the sector has been completely read, the floppy asserts the DONE line.

CHAPTER 5

DP278 COMMUNICATION OPTION

5.1 GENERAL

The DP278 communication option interfaces with the VT278 video terminal through two communications ports. Each port is individually programmable and allows for full modem control (DEC STD 052). The option contains firmware in the form of PROMs for terminal emulation. Each port is independent and does not present line speed loading to the other port.

The DP278 is available in the following two versions.

- **DP278-A** is the basic two-port communication option. It permits asynchronous and byte synchronous operation.
- **DP278-B** is similar to the DP278-A but contains additional circuitry for bit synchronous (BITSTUFF) operation.

5.2 PHYSICAL DESCRIPTION

The DP278 is an extended quad printed circuit board which fits into the VT278 card cage next to the CPU board. The CPU and DP278 modules are connected by an I/O connector board. Two 25-pin EIA connectors protrude from the back of the DP278 module and fit into the back shell of the VT278 casing. Most signals on the DP278 are generated by the CPU. Communication between the two modules is through IOT commands and interrupts.

5.2.1 Interrupts

There are three types of interrupts generated by the DP278: receiver interrupts, transmitter interrupts, and a modem change interrupt. Each type can be vectored to determine the cause of the interrupt.

5.2.1.1 Receiver Interrupts (2) – Each port can generate a receiver interrupt, if enabled, when a new character has been assembled from the communications line, or a parity CRC or overrun error has been detected (DP278-B only).

5.2.1.2 Transmitter Interrupts(2) – Each port can generate a transmitter interrupt, if enabled, when a new character can be loaded for transmission, or in case of transmitter underflow (DP278-B only).

5.2.1.3 Modem Change Interrupt – When either port detects a change of state on any incoming modem control line, if enabled, the modem change interrupt flags the change of modem state. The CPU can at any time issue an IOT to examine the modem control register contents of either port.

5.3 PORT DESCRIPTION

The majority of the hardware in the DP278-A composes the two identical ports. Each port consists of the following.

- Standard communication device (SCD) (contains circuitry standard asynchronous/synchronous data communication)
- Port status register
- Port control register
- Port modem control register

The SCD contains internal registers which can be accessed under program control. These registers are listed below.

- SYN1, SYN2, or DLE (write only)
- Internal status (read only)
- Mode register 1 (read/write)
- Mode register 2 (read/write)
- Command (read/write)

For more detailed information, refer to the *DECmate System Architecture Manual*.

The DP278-B consists of the above circuitry plus an extended synchronous device (ESD) for each port. The ESD contains circuitry to perform bit synchronous operation.

The following parameters may be chosen for either port.

Character length – 5, 6, 7, or 8-bit

Number of stop bits – 1, 1.5, or 2

Parity – odd, even, or none

Operating mode – half-duplex or full-duplex

Clocking – synchronous or asynchronous

Maintenance – local or remote loopback

Protocol – BITSTUFF (DP278-B only) or byte oriented

Line speed (baud) – 50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600, 19200

The ESD contains internal registers which can be accessed under program control. These registers are listed below.

- Receive data/status (read only)
- Transmitter data/status (read/write)
- Parameter control sync/address (PCSAR) (read/write)
- Data length select (PCSCR) (read/write)

For more detailed information, refer to the *DECmate System Architecture Manual*.

5.3.1 Standard Communication Device (SCD)

The SCD can be programmed to transmit or receive synchronous or asynchronous data in full-duplex or half-duplex operation.

Transmitted data is accepted from the VT278 DX bus and serialized for transmission. Received data is presented to the DX bus in an 8-bit parallel, right-justified (LSB) format.

5.3.1.1 Asynchronous Operation – Under asynchronous operation, the SCD can be programmed to do the following.

- Transmit or receive 5, 6, 7, or 8-bit characters
- Insert 1, 1.5, or 2 stop bits (dependent on character length)
- Compute and insert a parity bit
- Detect parity, framing, and overrun errors

5.3.1.2 Synchronous Operation – Under synchronous operation, the SCD can be programmed to do the following.

- Transmit or receive 5, 6, 7, or 8-bit characters
- Recognize and synchronize one or two sync characters
- Perform automatic SYNC or SYNC-DLE insertion, and SYNC or DLE character stripping, and allow for automatic SYN-SYN or DLE-SYN character fill

5.3.1.3 Clocking – Both the transmitter and receiver clock can be externally or internally generated by the SCD. Internal clocking is used for asynchronous operation. Clocking for synchronous operation is derived from the incoming modem signals, or the DP278 provides transmit timing to the modem. When the system is in maintenance mode, the external loopback maintenance clock speed is determined by the SLU speed on the CPU.

5.3.1.4 Maintenance – Internal maintenance checks (local and remote loopback) can be performed by the SCD. Local loopback causes the SCD to connect internally its transmitter output to receiver input and thus check for parity, status, overrun, and framing errors. Remote loopback allows a remote modem to send data to the SCD and have that data transmitted back without sending the data to the DX bus. Error flags due to transmission or reception errors are sent to the VT278. When in the maintenance mode, the transmitter serial output is held in the MARKing state.

5.3.1.5 Unused Pins – The SCD has integral modem control pins tied to appropriate logic levels (true) so that modem control is entirely under software control and is independent of the communications device. These unused modem control pins are listed below.

- DSR – data set ready (tied to logic 0)
- DTR – data terminal ready
- RTS – request to send
- CTS – clear to send (tied to logic 0)
- DCD – data carrier detected (tied to logic 0)

5.3.2 Extended Synchronous Device (ESD)

The ESD operates in both bit synchronous and character-oriented modes, thus allowing the choice of using only the SCDs or both the SCDs and ESDs. The ESD can transmit or receive 1 through 8-bit characters in bit synchronous mode, or 5 through 8-bit characters in character-oriented mode.

The addition of ESD offers the user more communications flexibility and a decrease in software overhead. This software savings is achieved because of the onboard error flagging available from the ESD, and its ability to perform protocol handling in hardware (including CRC generation and checking).

5.3.2.1 Clocking – For standard communications needs, all clocking information for the ESD is derived from the incoming modem lines. Under Maintenance Select (MSEL) however, the VT278 SLU 1 clock is substituted for the TXCLK (transmitter clock) input of the ESD.

5.3.2.2 Maintenance – When MSEL (Maintenance Select) is asserted on the ESD, a local loopback mode similar to that of the SCD is entered. When in the maintenance mode, the transmitter serial output is held in the MARKing state.

5.3.3 Port Control Registers

Each port contains an 8-bit write-only port control register. This register can be dynamically changed under program control to perform the following functions.

- Output modem control signals
- Determine which communications chip (SCD or ESD) is active in the port
- Force the maintenance state if ESD is active
- Multiplex the CPU SLU 1 clock for diagnostics
- Activate the transmit and receive portions of ESD (if present)

5.3.4 Port Status Registers

Each port contains a 12-bit read-only port status register. This register determines the status of the communication chips within the port. AC(10) determines the presence of an ESD in the port.

5.3.5 Modem Control

Table 5-1 defines modem signals supported by the DP278. The table lists the signal names of the various communications standards. The EIA RS-449 standard is used in this manual. The DP278 can be configured to include various modems. It may be necessary to set certain DIP switches and install or remove certain jumpers. Refer to the *DECmate Pocket Service Guide* (EK-VT278-PS) for specific details.

5.3.6 Line Handling

Outgoing modem control signals RS, TR, and SR are made available from the port control registers. These three signals, along with the incoming modem control signals (CS, DM, RR, SQ, and IC) are available to the DX bus from the port modem control registers (Table 5-1). It is therefore possible to write out RS, for example, and then read that output modem signal from the port modem control register.

Lines ST, RT, SQ, SR and TT can be switched to open circuit so that these drivers and receivers cannot affect or be affected by a modem. Furthermore, lines RL and LL can be jumpered to open circuit or EIA/CCITT off; they cannot be controlled by software. Protective ground can be switched to open circuit by a jumper.

Each outbound modem line is set to EIA RS-232-C/RS-423/CCITT V.28/V.10 equivalent levels. Likewise, inbound modem lines are shifted from EIA to TTL levels. Line slew rates for control lines (not-timing and data) are controlled to 100 microseconds. SD and TT default to RS-232-C/V.28 timing (1 microsecond). However, the slew rates for these lines can be controlled to meet RS-423 and CCITT V.10 by substituting a resistor for a jumper in a given port.

Table 5-1 Modem Signal Definitions

Pin No	Function	EIA RS-449	Circuit DIN 66020	EIA RS-232-C	CCITT V.23/V.24
1	Protective ground		E1	AA	101
2	Transmitted data	SD	D1	BA	103
3	Received data	RD	D2	BB	104
4	Request to send	RS	S2	CA	105
5	Ready for sending	CS	M2	CB	106
6	Data set ready	DM	M1	CC	107
7	Signal ground	SG	E2	AB	102
8	Data channel received line signal detector	RR	M5	CF	109
15	Transmitter signal element timing	ST	T2	DB	114
17	Receiver signal element timing	RT	T4	DD	115
18	Local loopback	LL	PS3		141
20	Data terminal ready	TR	S1.2	CD	108.2
21	Signal quality	SQ		CG	110
21	Remote loopback	RL	PS2		140
22	Calling indicator	IC	M3	CE	125
23	Data signaling rate selector	SR	S4	CH	111
24	Select standby	SS	S8		116
24	Terminal timing	TT	T1		113

5.3.7 Modem Interrupt Handling

Each port may interrupt the VT278 if a state change occurs on any incoming control line (CS, DM, RR, SQ, or IC). There is no interrupt priority. When any or all of these lines change state while the modem control interrupt is enabled, the VT278 is flagged and the applications program must examine both port modem control registers to determine the new state of the lines. Note that any of ten incoming modem control lines could have caused the interrupt.

5.3.8 Slew Rate

For all standards (except CCITT V.10), a 10K ohm slew rate resistor is included on the DP278. If a modem conforms to CCITT V.10, a different slew rate resistor is necessary. The resistor value depends on the baud rate. Refer to the *DECmate Pocket Service Guide* (EK-VT278-PS) for specific details.

5.3.9 I/O Structure

Devices 30 and 34 read the internal registers of the SCDs and ESDs. Devices 31 and 35 write the internal registers of these controllers.

Device Code	Function
30	Port 0 read
31	Port 0 write
34	Port 1 read
35	Port 1 write
36	Control

Device 36, used for control, is explicitly decoded for the following functions.

IOT Code	Function
6362	Write port 0 control register
6363	Read port X status register*
6364	Read port X modem control register*
6366	Write port 1 control register
6367	Reset communications device

5.3.10 Terminal Emulation Hardware

The DP278 communication firmware is a 1019-word, 278 assembly language program that allows the VT278 terminal to emulate a "dumb" terminal and accept a dual-line load from a host computer.

This firmware is stored in sets of 1K by 4 PROMs which are inserted into sockets on the M8437 module. The PROMs are addressed by auto-sequencing counters on the module, and appear as a 1K portion of control panel memory field 5 (Figure 2-5). Firmware is loaded into main memory (field 3) when either KEY 4 or KEY 5 are found to be pressed on the VT278 keyboard in set-up mode. When this condition is found, the code is read from the PROMs by power strobing them and placing their contents in main memory, where execution of that code begins. The address counters are reset at the end of the VT278 self test, and at the start of the KEY 4 down and KEY 5 down firmware routines.

*X may be 0 or 1.

5.3.11 Power Consumption

The power consumption of the DP278 is as follows.

DP278-A

Voltage	Current	
+5 Vdc	650 mA	(3.25 W)
+12 Vdc	175 mA	(2.1 W)
-12 Vdc	175 mA	(2.1 W)
		<hr/>
		7.45 W total

DP278-B

Voltage	Current	
+5 Vdc	950 mA	(4.75 W)
+12 Vdc	375 mA	(4.5 W)
-12 Vdc	175 mA	(2.1 W)
		<hr/>
		11.35 W total

5.3.12 Switches and Jumpers

Table 5-2 lists switches and jumpers that make or break the connection between EIA ports J1 and J2 and the rest of the VT278.

5.4 FUNCTIONAL BLOCK DIAGRAM

Figure 5-1 is a functional block diagram of the DP278 communication option. The DP278 interfaces to the CPU through the DX bus.

The DX bus connects to the finger pins of the CPU board. This 12-bit bus contains a buffered portion (BDX) and a latched portion (LDX). The I/O controller (IOC) chip is a read/write device directly connected to the DX bus. The I/O controller monitors flags received from devices that cause an interrupt. These include port 0 and port 1 receive flags, port 0 and port 1 transmit flags, and the modem flag.

The COM PRES line indicates to the CPU that the DP278 is present. The priority output (PRO) line from the DP278 indicates that the I/O controller on the DP278 has been programmed.

5.4.1 BDX Bus

The BDX bus is a buffered portion of the DX bus. It is only possible to write to the BDX bus. The following devices are connected to it.

- Address latch (BDX bits 0 – 3)
- Device 36 decoder (BDX bits 9 – 11)
- GODE decoder (BDX bits 9 – 11)
- Port 0 control register (BDX bits 4 – 11)
- Port 1 control register (BDX Bits 4 – 11)
- LDX latch (BDX Bits 0 – 11)

The address latch receives data from the BDX bus and stores bits 0 through 3 of the DX bus during the IOT write time.

Table 5-2 Switch and Jumper Configurations

	J1 J2 Pin No	Function	Circuit EIA RS-449	DIN 66020	EIA RS-232-C	CCITT V.24
J1-W8	1	Protective ground		E1	AA	101
J2-W12	1	Protective ground		E1	AA	101
J1-S1	15	Transmitter signal element timing	ST	T2	DB	114
J2-S6	15	Transmitter signal element timing	ST	T2	DB	114
J1-S2	17	Receiver signal element timing	RT	T4	DD	115
J2-S7	17	Receiver signal element timing	RT	T4	DD	115
J1-W6	18	Local loopback	LL	PS3		141
J2-W10	18	Local loopback	LL	PS3		141
J1-S3	21	Remote loopback	SQ		CG	110
J1-W5	21	Remote loopback	RL	PS2		140
J2-S8	21	Signal quality	SQ		CG	110

Device 36 decodes bits 9 through 11 of the BDX bus during the 6362, 6363, 6364, 6366, or 6367 IOTs. When device 36 is enabled by the I/O controller, the last three bits (bits 9 – 11) of the BDX bus are decoded to determine the least significant octal digit of the 636X IOT (where X is the least significant octal digit).

GODE (general option device enable) decodes the 6134 IOT generated on the CPU module. The accumulator is monitored during the IOT write time to determine if the system responds to the IOT. For example, the DP278 has an accumulator value of 1. When a 1 is recognized by the DP278, during 6134 IOT, the accumulator is cleared. This informs the CPU that the DP278 board is connected to the system and is available to provide a boot PROM.

The port 0 and port 1 control registers are identical write-only registers. Bits 4 through 11 of the BDX bus are written. The 6362 IOT causes a write to the port 0 control register through the BDX bus and the 6366 IOT causes a write to the port 1 control register through the BDX bus.

The BDX bus constantly feeds the LDX latch, which stores data for the LDX bus during I/O write time. Thus, to write a device on the LDX bus, it is latched in the LDX latch by the CTL IOW signal. The tri-state LDX latch is enabled only during IOT write time. For an IOT read cycle, the data con-

Table 5-2 Switch and Jumper Configurations (Cont)

	J1 J2 Pin No	Function	Circuit EIA RS-449	DIN 66020	EIA RS-232-C	CCITT V.24
J2-W9	21	Remote loopback	RL	PS2		140
J1-S5	23	Data signaling rate selector	SR	S4	CH	111
J2-S10	23	Data signaling rate selector	SR	S4	CH	111
J1-S4	24	Select standby	SS	S8		116
J1-S4	24	Terminal timing	TT	T1		113
J2-S9	24	Select standby	SS	S8		116
J2-S9	24	Terminal timing	TT	T1		113

NOTE

For extended RS-232-C operations, set the following switches to the closed (on) position: S1, S2, S4-S7, S9, S10. Set S3 and S8 to open (off). Also install jumpers W7, W8, W11, and W12.

W13 and W14 are fixed non-optional jumpers.

To run DP278 diagnostics with external loopback, close (set to on) all switches (S1-S10). Be sure that W5 and W9 are not installed.

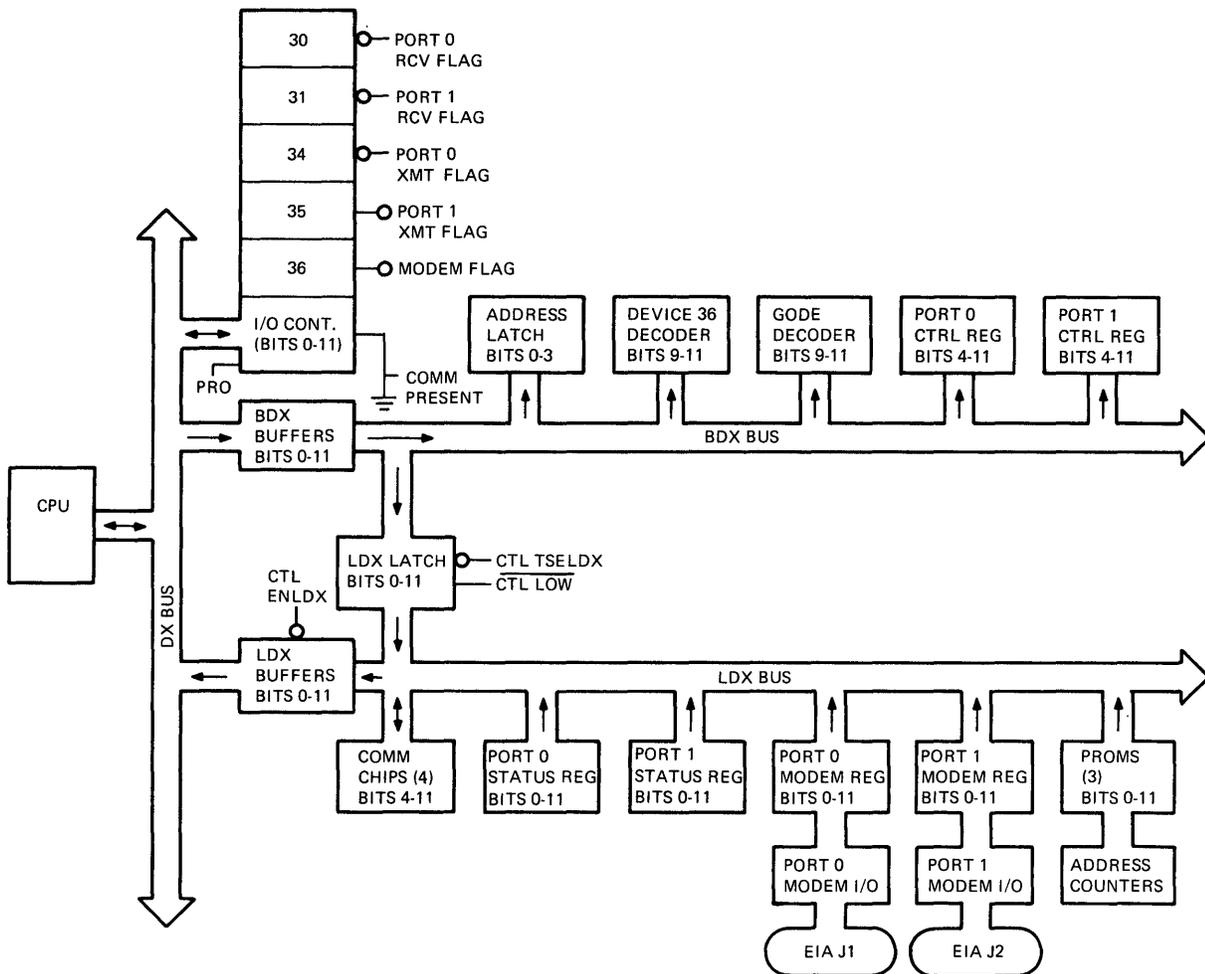
tained in the LDX latch is not driven onto the LDX bus since the bus may be servicing other devices (CTL TSELDX not asserted). When CTL TSELDX is asserted, the data stored in the LDX latches is enabled onto the LDX bus.

5.4.2 LDX Bus

The LDX buffers are enabled by the assertion of CTL ENLDX, and they gate the contents of the LDX bus onto the DX bus. The LDX, or latched portion of the DX bus, is connected to the following devices.

- Two SCDs (port 0 and port 1)
- Two ESDs (port 0 and port 1)
- Port 0 and port 1 status registers
- Port 0 and port 1 modem registers
- Three PROMs

The SCDs and ESDs are the only read/write elements on the LDX bus. The port 0 and port 1 status registers are read by the 6363 IOT. The register to be read is determined by the state of accumulator bit 0 at the time the IOT is issued. If bit 0 is 0, the 6363 IOT reads the port 0 status register. If bit 0 is 1, the 6363 IOT reads the port 1 status register.



MA-718B

Figure 5-1 DP278 Functional Block Diagram

The port 0 and port 1 modem registers are read by the 6364 IOT. The register to be read is determined by the state of accumulator bit 0 before the read. If bit 0 is 0, the 6364 IOT reads the port 0 modem register. If bit 0 is 1, the 6364 IOT reads the port 1 modem register.

The three PROMs form a 12-bit read-only memory used for booting the system into terminal emulation mode. LDX buffers are enabled onto the DX bus during the DPR to ROMs cycle.

LDX buffers enable the contents of the LDX bus onto the DX bus when any of the following conditions are met.

- Reading from the DP278 through IOT 6304, 6306, 6344, 6346, 6363, 6364 (device code 30 or 34) in the DP278
- Executing IOT 6363 or 6364
- Reading the ROMs

The blocks on Figure 5-1 labeled port 0 and port 1 modem I/O are the EIA line drivers and receivers used for transmission onto the LDX bus. The line drivers receive information from the SCDs, ESDs, and port control registers. The line receivers receive information from the EIA connectors and transmit it to the modem register, SCDs, ESDs, and modem interrupt logic in the modem I/O block. EIA J1 is a 25-pin connector connected to the port 0 modem I/O. EIA J2 is a 25-pin connector connected to the port 1 modem I/O. Address counters connected to the PROMs form the PROM addresses. These counters are cleared by the I/O CLR line which is triggered by the CAF instruction. The counters also can be cleared by issuing IOT 6366. This IOT serves a dual function: it writes the port 1 control register and clears the address counters.

(

(

)

CHAPTER 6 RL278 DISK CONTROLLER OPTION

6.1 GENERAL

The RL278 disk controller option is a random access, mass storage device that stores up to 10.48 million bytes or 6.96 million 12-bit words (per disk) in fixed-length blocks on preformatted disk cartridges. Transfers are programmable for 8-bits or 12-bits. The RL278 controller provides interface and control functions between the DX bus and RL02 disk drives.

The RL278 interface and control logic is contained on a single multilayer 10.5 in × 11.1 in module (M8439). It is inserted into the VT278 backplane and connected to the first drive by a BC26T interface cable. Connections between the controller and other drives are made using the conventional daisy chain method with a 7012122 cable.

The RL278 controller can control up to four RL02 disk drives (Figure 6-1). It communicates with one or more drives, but can only transfer data to or from one drive at a time. Data transfers to and from memory are accomplished by the single cycle data break (DMA) facility.

The RL02 disk drive is a random access, moving head, dual-surface, single-platter drive. Each drive has two movable read/write heads, which use the RL02 disk cartridges as storage medium. The RL02 employs the servo-in-data concept. Thus, head positioning and track counting information are derived from pulses prerecorded in the data record. In effect, each read/write head seeking a track becomes its own servo transducer. Therefore, an alignment cartridge is not required for head alignments.

The disk drive consists of a spindle motor, head positioner, drive electronics, power supply, cabinet, and front panel assembly. The storage medium for the drive is a top-loading, single-disk cartridge.

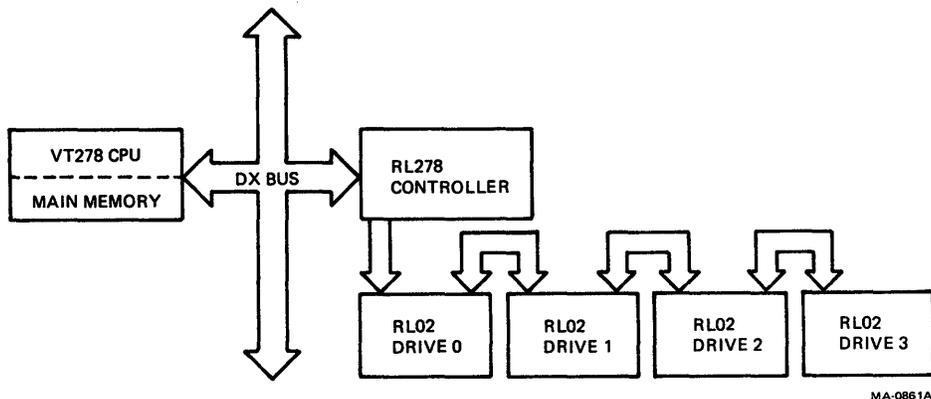


Figure 6-1 RL278 Controller Block Diagram

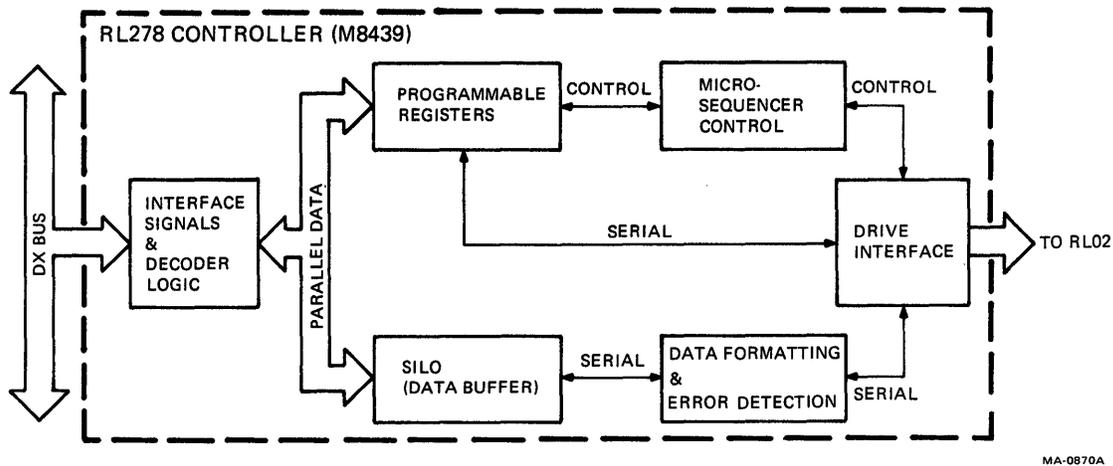


Figure 6-2 RL278 Controller Functional Block Diagram

6.2 FUNCTIONAL DESCRIPTION

Figure 6-2 shows the controller interface between the DX bus and RL02 disk drives. The controller contains function and format logic for controlling data transfers between itself and the processor. It also contains logic for receiving and transmitting commands and signals between the CPU and the drive(s).

The RL278 controller performs four basic functions.

1. Positions read/write heads
2. Controls data flow between DX bus and drive(s)
3. Arbitrates status information
4. Detects subsystem (disk and controller) errors

These four capabilities, taken as a group, can be called the function and format control logic. Function control logic decodes the operation to be performed and then conditions the format control to execute the function. Format control uses binary counters to address a ROM that provides the necessary output control signals in the proper sequence for command execution.

In a typical read/write application, function control starts to count, sets up various control conditions, reaches a predetermined state, and waits. Format control then takes over. It establishes necessary word boundaries, controls the actual transfer of header information and data, and then reverts operation back to function control for command completion. Format control is operative only during execution of those commands associated with head positioning and/or the actual transfer of header information or read/write data, in synchronization with the disk drive.

In addition to the standard interface logic, the RL278 controller contains.

- First-in, first-out 16-word \times 12-bit silo and bus drivers
- Write precompensation circuit for offsetting the effects of peak shifting normally associated with modified frequency modulation (MFM) recording
- Phase lock loop circuit for compensating variations in disk speed

- Data separation circuit for separating clock and NRZ data signals from drive MFM encoded serial data
- Header compare circuit for searching headers on recording surfaces
- Cyclic redundancy check (CRC) circuit for detecting header and data recording errors

6.2.1 Operation

The following paragraphs describe the sequence of events during execution of all controller commands. After the necessary registers are loaded, the command is decoded and function control is initiated. Controller logic and error bits are cleared. The command function is then executed.

When the command function is completed, the Done flag and interrupt enable are checked. If the interrupt is not enabled, a skip function is generated by setting the Done flag and the command is terminated. If the interrupt is enabled, it causes the CPU to jump (JMS) into an interrupt service routine (location 0).

To issue a function command, the CPU places the address and data onto the DX bus. The controller decodes the information and channels it to the appropriate register. When the function command is written into command register B, it initiates the microsequencer routine. The microsequencer control performs a different routine for each controller command function. These routines manipulate data formatting circuits in order to format the data. The data formatting function includes an error detection feature that uses cyclic redundancy checking (CRC).

Before reading or writing data, it is necessary to verify that the drive heads are positioned properly on the disk. First, execute a read header command to determine where the heads are located, and whether the upper or lower head has been selected. If heads are positioned at desired track address, the read/write command can be executed directly. If not, a software calculation must be performed to determine the cylinder distance that the heads must be moved to reach the proper location. This cylinder distance information is transmitted to the selected drive by a seek command.

6.3 COMMAND EXECUTION

Eight controller commands are embedded in command register B (bits 9–11). Three of these are used for initialization and test purposes. The remaining commands control head positioning and the actual transfer of header information and read/write data between the controller and disk drive(s).

**Installation and
Test Commands**
Maintenance
Reset
Get Status

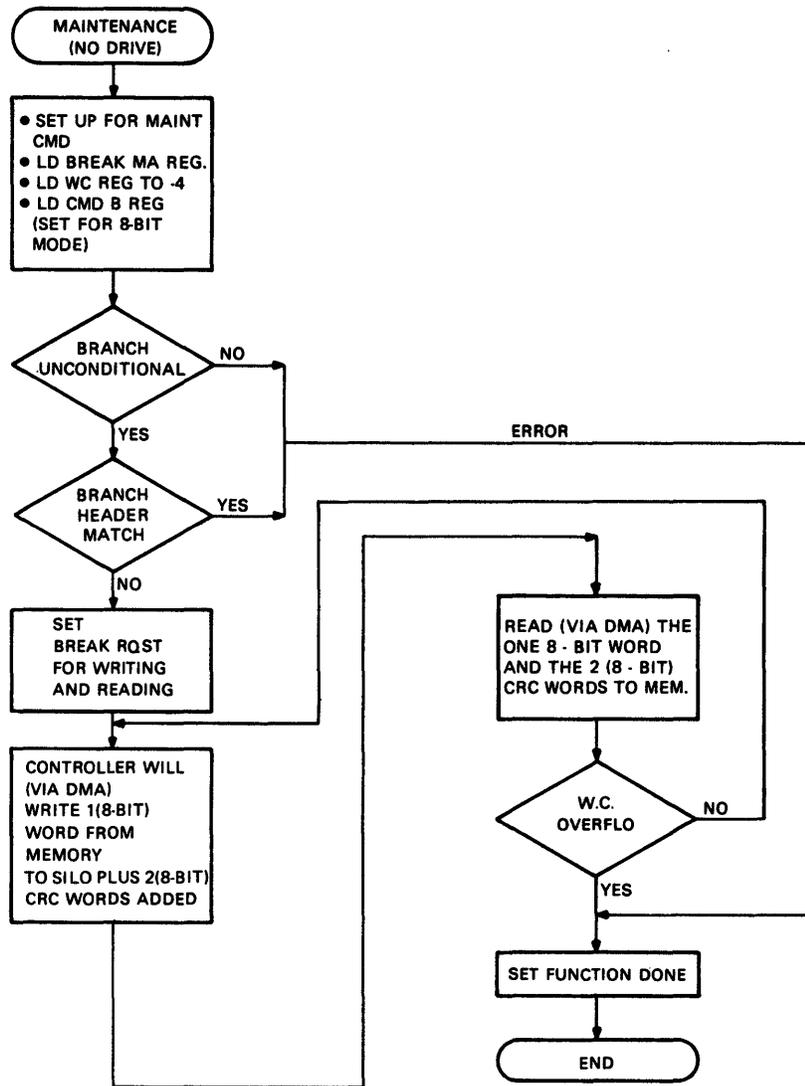
**Transfer Header and
Read/Write Commands**
Seek
Read Header
Write Data
Read Data
Read Data (without header check)

6.3.1 Maintenance

The maintenance command exercises controller logic to ensure that it is operating properly (Figure 6-3). It tests command registers, silo, and data paths without the disk drive connected.

When the MAINT command is decoded the following sequence occurs.

1. One DMA occurs out of memory at the location specified by the break MA. A write data function is executed out of the silo.



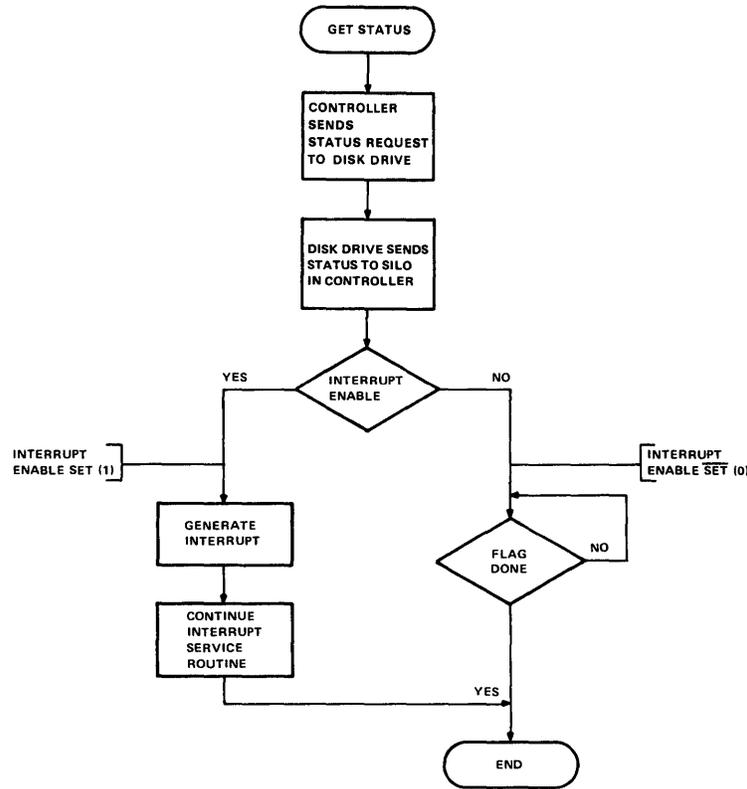
MA-1429

Figure 6-3 Maintenance Command

2. Written data is loaded into the silo, along with two 8-bit words (CRC character).
3. DMAs cause the data (8-bit) word and the two CRC words to be written into memory.
4. This sequence repeats until word count overflow. Then, a function done interrupt occurs. A CRC error and function done immediately raises an interrupt if a CRC error occurs.

6.3.2 Reset

This command resets all error bits in the drive. It does not clear any registers in the controller, nor does it cause heads to move to track 0.



MA-0885

Figure 6-4 Get Status Command

6.3.3 Get Status

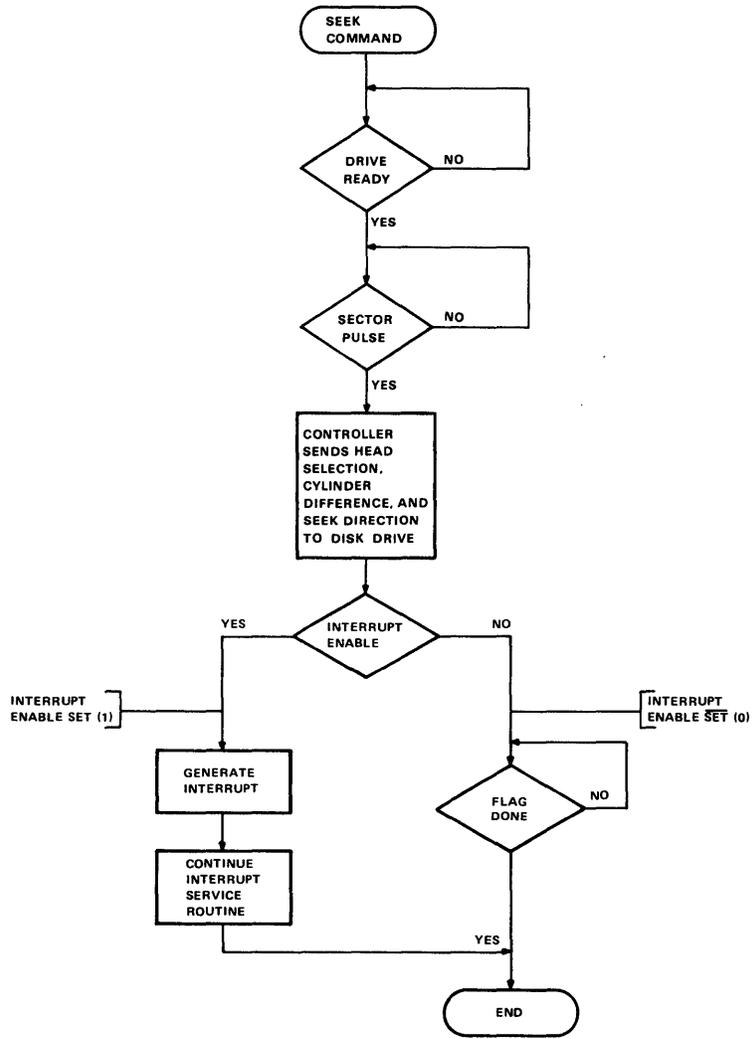
Figure 6-4 shows a status request sent to the disk drive through a serial drive command line. The disk drive returns its status through a serial status line. Status information is loaded into the silo and the function is terminated. Drive status can now be obtained by reading the silo. This command can be executed whenever the controller is ready, even though the drive may not be ready (for example, during a seek or when in the load state).

6.3.4 Seek

After the disk address register (DAR) is loaded, the seek command is decoded and function control is initiated (Figure 6-5). Various circuits and error bits are cleared. When the drive is ready to accept the seek command, bit 11 (drive ready) of the error register is set. The controller waits for a sector pulse from the drive, and then initiates format control. The 16-bit word in the disk address register is shifted out serially on the drive command line to the drive. When transfer is complete, format control is terminated, and control reverts back to function control. Operation is now the same as other commands.

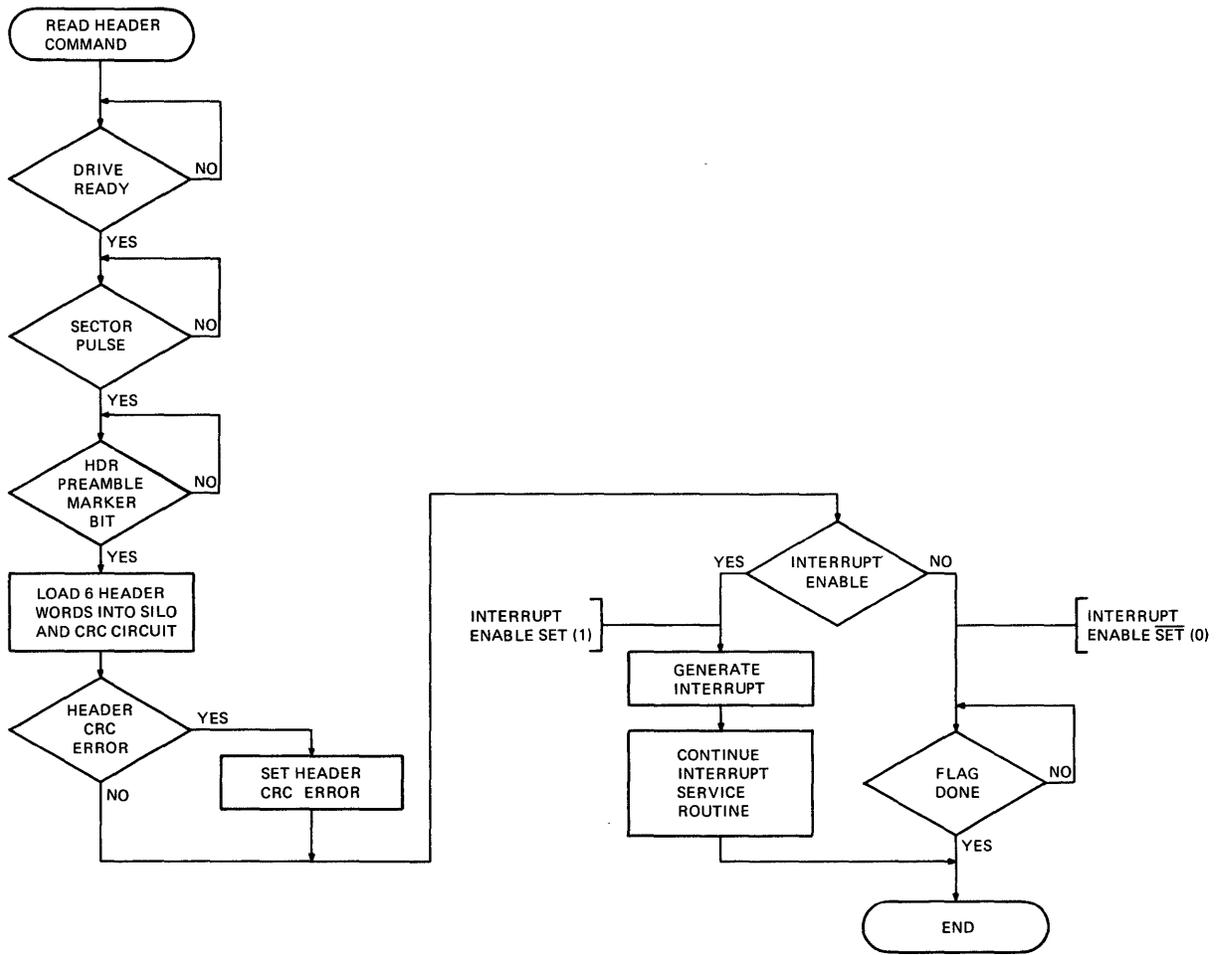
6.3.5 Read Header

When the read header command is decoded, function control initiates and various circuits and error bits are cleared (Figure 6-6). When the drive is ready to accept a command, drive ready sets. The controller waits for a sector pulse from the drive, and then initiates format control. Read circuits in the phase lock loop/data separator are enabled. When the header preamble marker bit is detected, the first header word is directed through the silo. At the same time, this first header word is loaded into the CRC circuit. This operation is repeated for the next two header words. Read circuitry is then disabled, and a header CRC error check is performed. If there is no header CRC error, function control takes over, and terminates the command.



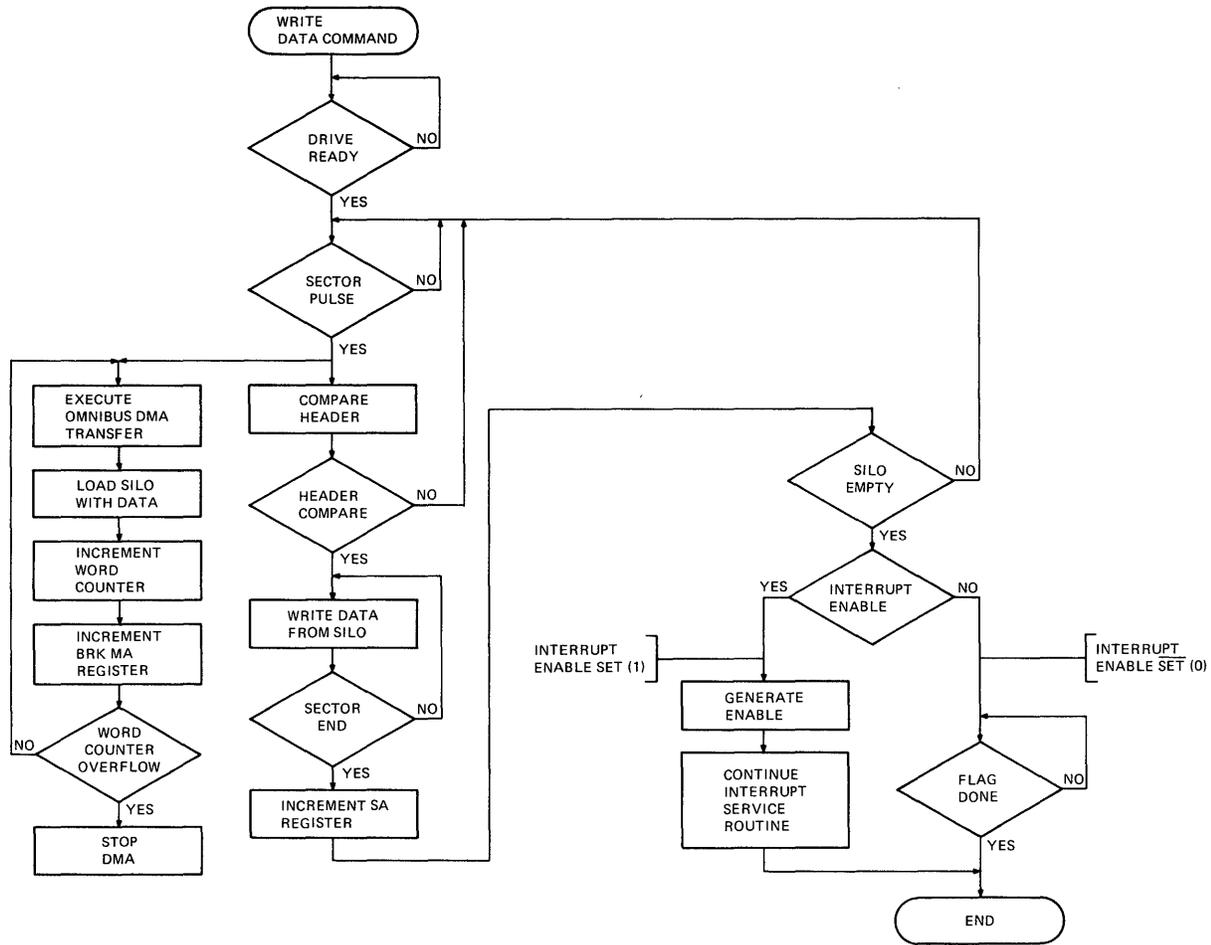
MA-0892

Figure 6-5 Seek Command



MA-0893

Figure 6-6 Read Header Command

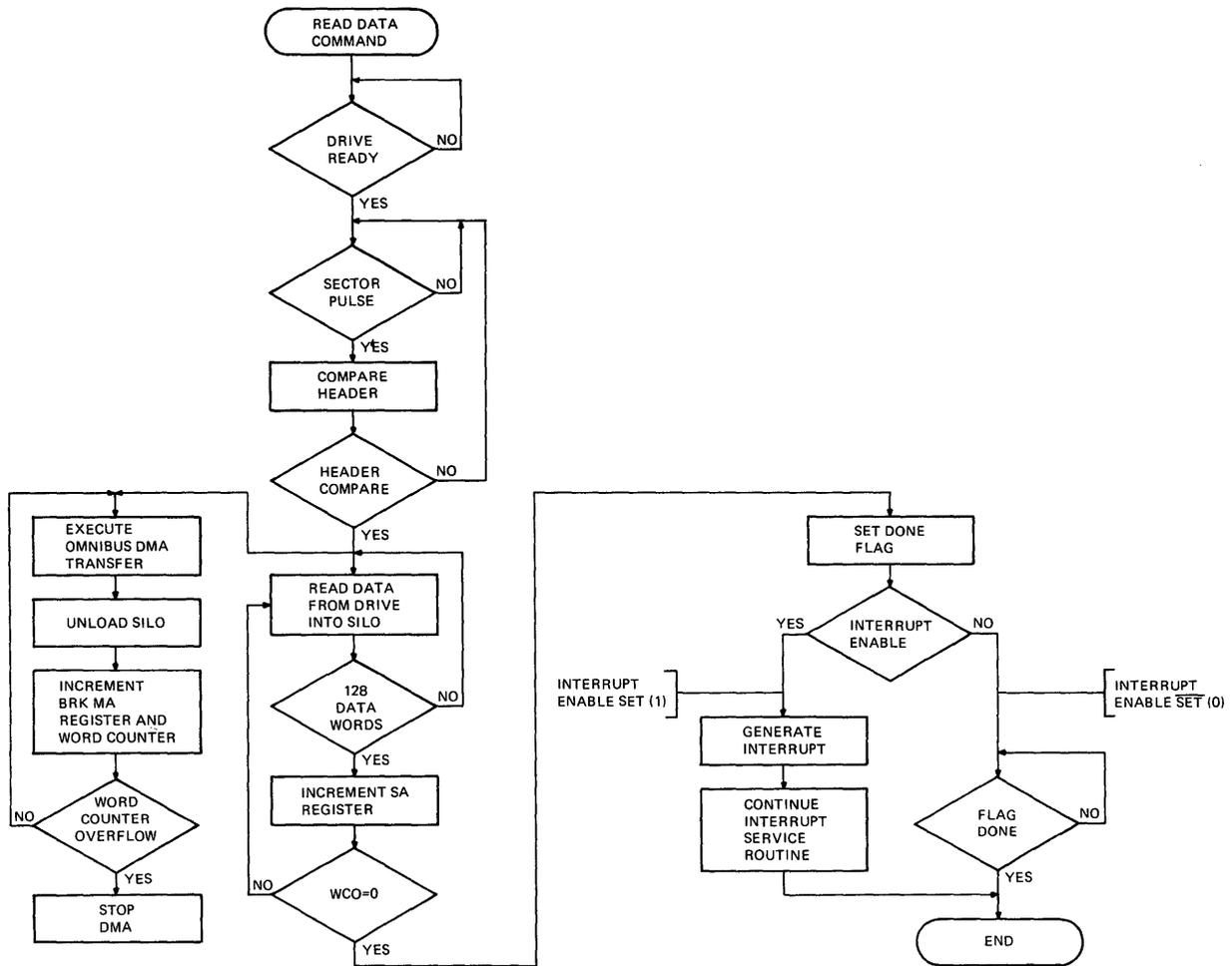


MA-0894

Figure 6-7 Write Data Command

6.3.6 Write Data

After the write data command is initiated (Figure 6-7), DMA cycles are immediately executed on the DX bus. Parallel data words are loaded into the silo. At the same time, format control circuit performs a search for the correct header. After successful header comparison, writing to the disk drive begins. Before it reaches the disk drive, serial data is shifted out from the silo to the write data encoding circuit.

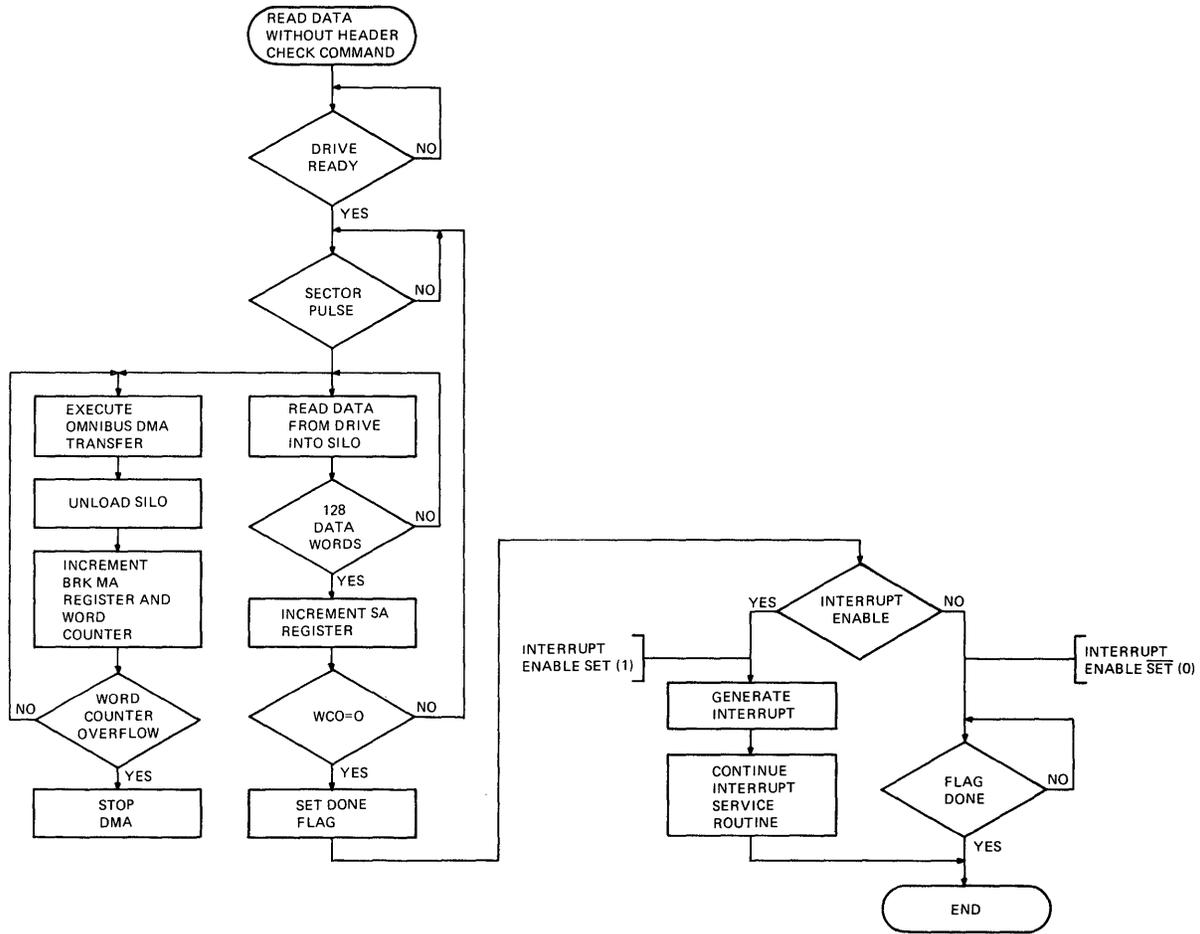


MA-0872

Figure 6-8 Read Data Command

6.3.7 Read Data

After the read data command is initiated (Figure 6-8), a search for the correct header is performed by format control circuit. After successful header comparison, DMA cycles are immediately executed on the DX bus. Parallel data is transferred from the silo to the DX bus.



MA-0895

Figure 6-9 Read Data (Without Header Check) Command

6.3.8 Read Data (Without Header Check)

This command is the same as a read data command, except that no header comparison occurs before reading data from the disk drive into the silo (Figure 6-9). The first sector encountered is read and shifted into the silo. There is no check for a header CRC error. However, data is checked for a CRC error.

Digital Equipment Corporation