EK-VT278-HR-001

DECmate

System Architecture Manual

digital equipment corporation • maynard, massachusetts

1st Edition, June 1981

Printed in U.S.A.

CONTENTS

FIGURES

TABLES

PREFACE

This is a reference manual for assembly language programming of the DECmate system. DECmate consists of the VT278 intelligent video terminal, and a number of optional devices. It is assumed that you already know programming and are familiar with the assembler available on the OS/278 operating system.

This manual is organized in a usable order. It is not necessary that you read the chapters sequentially. Each chapter and major section is independent from the others. This allows reference to specific information without frequent cross-referencing.

Chapter 1 provides general information about the DECmate system hardware and software.

Chapter 2 is an overview of the major DECmate functional circuits and options.

Chapter 3 describes the VT278 operating procedures which include self-test, SET-UP, and terminal mode communication procedures.

Chapter 4 describes the fundamental VT278 instruction set.

Chapter 5 contains the programming information for each of the VT278 standard devices. They include the keyboard/display, printer port, real-time clock, and the RX278 floppy disk subsystem interface.

Chapter 6 contains the programming information for the optional DP278 communications device.

Chapter 7 contains the programming information for the optional RL278 cartridge disk subsystem.

The appendices provide reference information such as ASCII and graphics character sets, foreign language keyboards, detailed system and option specifications, and programming considerations for programmers who are converting from PDP-8 to VT278 systems.

You may also need to refer to the following documents:

- DECmate Owner's Guide (AA-K33ØA-TA)
- DECmate Technical Description (EK-VT278-TD)
- DECmate Pocket Service Guide (EK-VT278-PS)
- DECmate Maintenance Print Set (MP00900)

CHAPTER 1
INTRODUCTION

1.1 DECmate GENERAL DESCRIPTION

DECmate is a modular computer system, designed for use in office environments. DECmate consists of a VT278 intelligent video terminal and a number of options, which include disk storage subsystems and communications facilities.

All DECmate functions are controlled by a single board processor mounted inside the terminal case. Figure 1-1 shows the major components of DECmate. The components have the following features:

- Central processor that executes an enhanced PDP-8A instruction set
- Main memory that contains 32K words of random access memory
- Control panel memory that supplies a second address space outside the DECmate's normal 32K words of main memory
- Integrated architecture that incorporates the keyboard and video display as closely coupled peripherals
- A standard serial printer port
- Up to four optional RXØ2 diskette drives
- Optional ports supporting two serial communication lines
- Up to four optional RLØ2 disk drives
- Real-time clock
- An automatic self-test procedure which is executed at power-up.

1.2 DECmate OPTIONS

DECmate supports five user options. Each option can be ordered by contacting your local DIGITAL Sales Office.

The DP278 communication option provides two programmable serial communication ports in addition to the printer port.

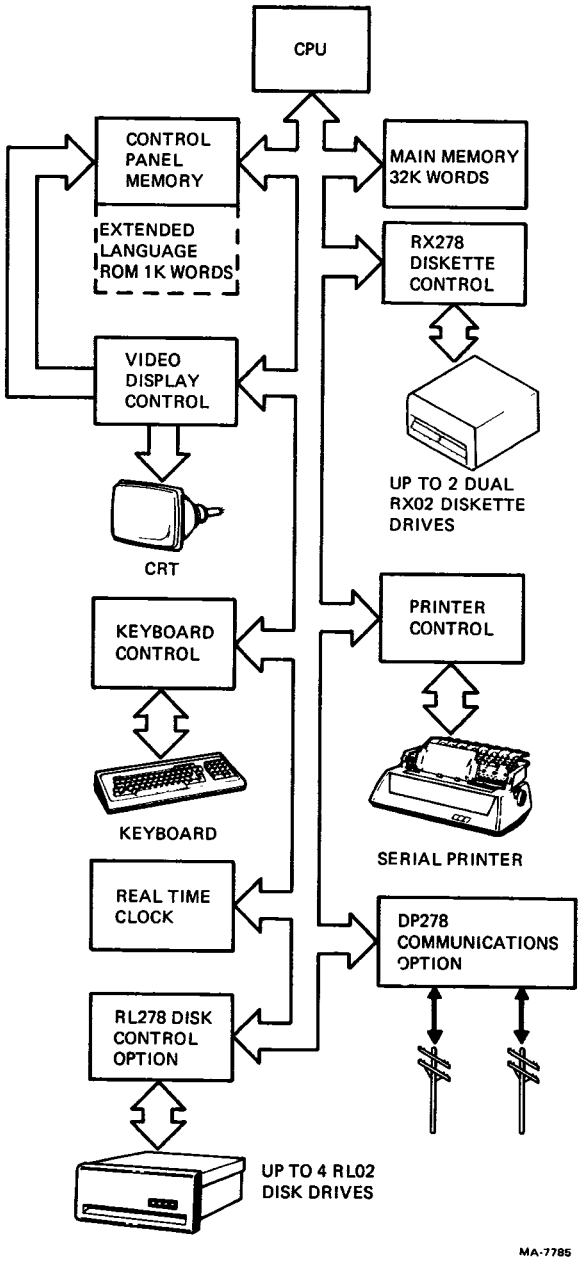


Figure 1-1 DECmate Major Components

The RL278 disk controller option connects up to four RLØ2 disk drives to the basic DECmate system. This option allows access to over 27 megabytes of additional mass storage.

The Extended Language ROM allows four foreign languages (in addition to the standard United States and United Kingdom English) to be displayed. Foreign language keyboards are available to support each language:

French
French-Canadian
Dutch
German.

The serial printer option can be connected to the printer port on the VT278.

The diskette drive option allows as many as four diskette drives to be connected to the diskette interface on the VT278.

1.3 DECmate OPERATING SYSTEMS

The three operating systems available from DIGITAL for DECmate are described in the following paragraphs.

1.3.1 WPS 8/278 Word Processing System

When used with word processing software DECmate supplies a hardware/software text processing system for office and business use. WPS 8/278 allows the user to:

- Prepare and edit reports which may require several drafts before final printing.
- Print form letters using a stored form document and a list from which items, such as names and addresses, are automatically selected.
- Send documents via an asynchronous, serial, RS232-C compatible communication line.

1.3.2 COS-310 Commercial Operating System

COS-310 is an applications development tool for EDP users who want to use data management functions for small to medium-size business applications. It provides the user with:

- An operation control monitor
- An easy-to-learn, high level programming language (DIBOL--Digital Equipment Corporation's Business Oriented Language)
- An interactive line-oriented text editor that is part of the monitor
- Program preparation, debugging, and production utilities.

1.3.3 OS/278 Operating System

OS/278 is a comprehensive executive for DECmate. It has extensive application software development tools and an efficient runtime environment for producing application programs. OS/278 is controlled through a concise command language (CCL) that simplifies program development and execution.

Programs stored on disks or diskettes are loaded, modified, or executed by simple keyboard commands. OS/278 also allows program chaining so that a complex program can be divided into a series of smaller modules.

1.4 SPECIFICATIONS

This section contains a summary of specifications for DECmate system components. Detailed specifications for each system device are contained in Appendix F.

1.4.1 VT278 Video Terminal

Dimensions	See Figure 1-2
Weight	
Monitor	13.6 kg (30 lbs)
Keyboard	2.0 kg (4.5 lbs)
Shipping weight	18.6 kg (41 lbs)
Display	12-inch CRT displays 7 X 9 dot matrix characters with descenders
Format	24 lines X 80 characters or 14 lines X 132 characters
Character set	96-character displayable ASCII subset (upper- and lowercase, numeric, and punctuation)
Cursor type	Blinking block character
Keyboard	
General	83-key detachable unit with 1.9 m (6 ft) coiled cord
Key layout	65-key arrangement similar to standard typewriter keyboard with 18-key auxiliary keypad
Auxiliary keypad	18-key numeric pad with period, comma, minus, enter, and four general purpose function keys
Visual indicators	Seven LEDs: two for on-line/off line; one for data set ready (DSR) (also user-programmable); four are user programmable
Audible signals	Keyclick: simulates typewriter bell; sounds upon receipt of BEL code
I/O Interconnections	
Floppy disk	37-pin female connector connects one or two dual drives

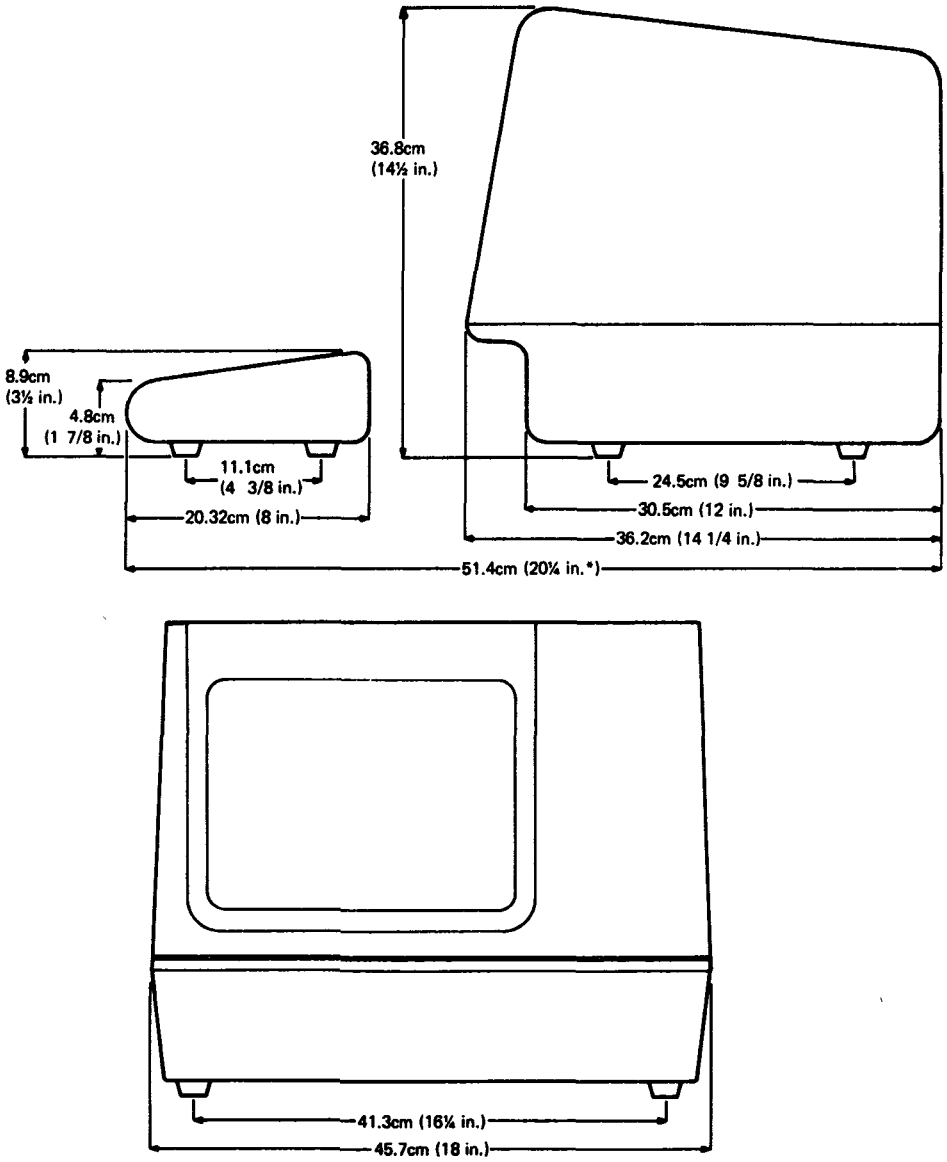


Figure 1-2 VT278 Dimensions

Printer	25-pin male connector
DP278 communications controller (optional)	Two 25-pin male connectors
RL278 disk controller (optional)	37-pin male connector

1.4.2 RX82 Floppy Diskette Drive

Dimensions	See Figure 1-3
Weight	50 lbs (approx.)

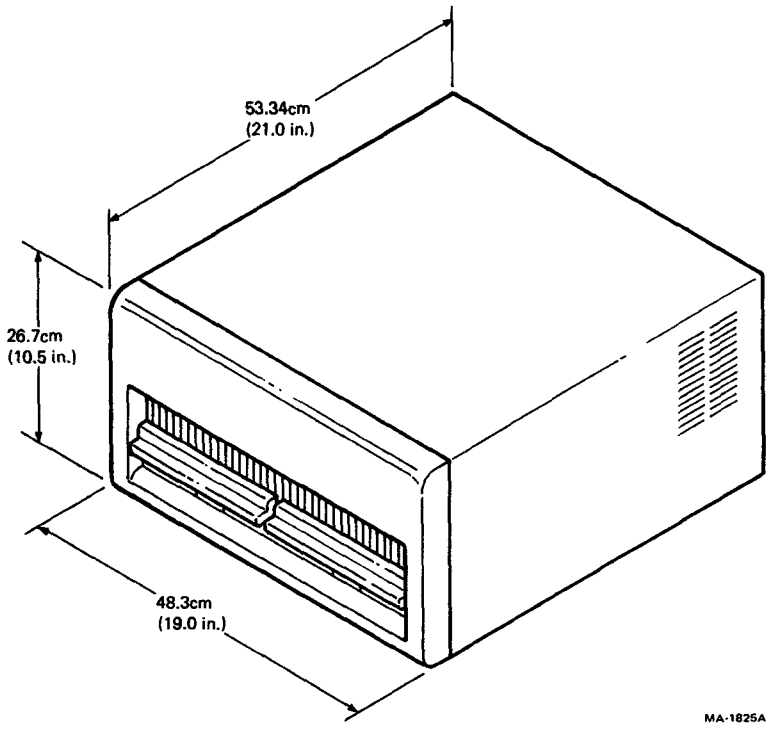


Figure 1-3 RXØ2 Diskette Drive Dimensions

Media	Each of two floppy disks supplies one surface with 77 tracks. Formatted capacity of 256,256 12-bit words per diskette.
Transfer rate	5ØØK bits/sec
Seek time	262 ms average

1.4.3 RLØ2 Cartridge Disk Drive

Dimensions	See Figure 1-4
Weight	34 kg (75 lbs)
Media	Disk cartridge supplies two surfaces with 512 tracks per surface. Formatted capacity of 1Ø.4 megabytes per cartridge.
Transfer rate	4.1 megabits/sec $\pm$1%
Seek time	55 ms average

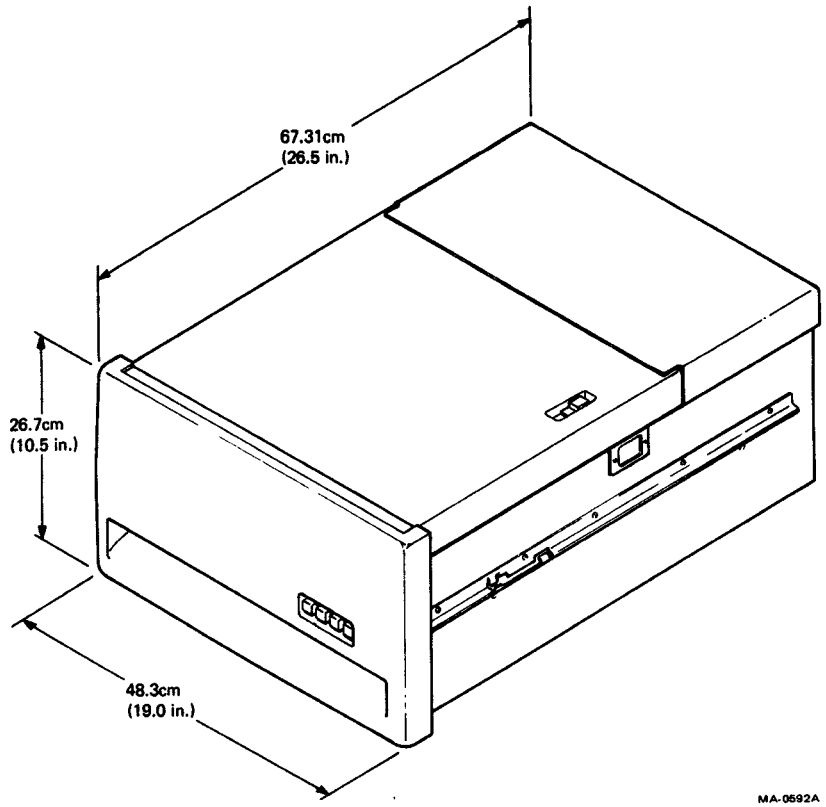


Figure 1-4 RLØ2 Dimensions

1.4.4 DECmate System

Environment

Operating

Temperature	15° to 32° C (59° to 9Ø° F)
Relative humidity	2Ø to 8Ø percent
Maximum wet bulb	25° C (77° F)
Minimum dew point	2° C (36° F)
Maximum altitude	2.4 km (8ØØØ ft)

Nonoperating

Temperature	-35° to +6Ø° C (-3Ø° to +14Ø° F)
Relative humidity	5 to 98 percent
Maximum altitude	9.1 km (3Ø,ØØØ ft)

Power

Line voltage	9Ø--128 V rms single phase, 3 wire or 18Ø--256 V rms single phase, 3 wire (switch-selectable)
Line frequency	47--63 Hz

CHAPTER 2
SYSTEM DESCRIPTION

2.1 GENERAL

This chapter provides a brief functional description of the DECmate system. More detailed information on DECmate and its associated components is found in the DECmate Technical Description and the DECmate Maintenance Print Set.

2.2 DECmate SYSTEM OVERVIEW

Figure 2-1 is a block diagram of DECmate. The following discussion describes each functional block of the diagram.

2.2.1 VT278 CPU Board

The CPU board controls all receiving, sending, and display functions within the terminal. It receives data from the:

Keyboard
RXØ2 diskette drives
Printer (XON-XOFF data control characters)
Communication lines if the DP278 option is installed
RLØ2 disk drives if the RL278 disk subsystem option is installed.

The CPU board sends command information and data to the:

Keyboard
Video monitor
Serial printer
RXØ2 Diskette drives
Communication lines if the DP278 option is installed
RLØ2 disk drives if the RL278 option is installed.

The CPU board also has two LEDS:

1. PWR OK lights to indicate that the dc voltages are present and within tolerance

2. CPU OK blinks at a 1 Hz rate to indicate that the CPU is functional.

The CPU board contains the following major circuits. The circuits are described in the following paragraphs.

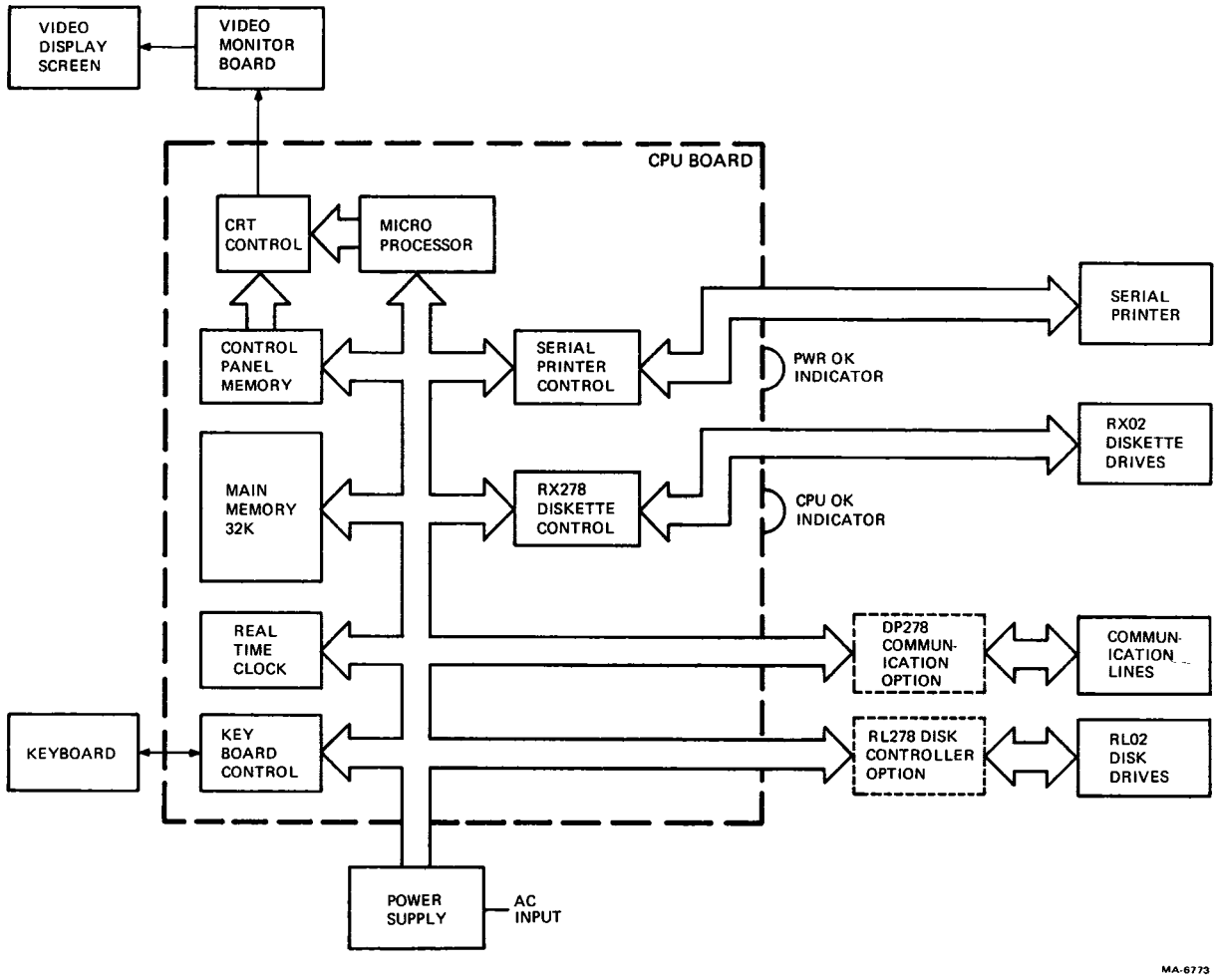


Figure 2-1 DECmate Block Diagram

Microprocessor
Control panel memory
Main memory
CRT control
Real-time clock
RX278 diskette subsystem control
Printer port

2.2.1.1 Microprocessor -- The 6120 microprocessor is a high speed CMOS processor that executes the VT278 instruction set. The microprocessor has a dual purpose in the VT278:

1. It executes the user program from the 32K word main memory.

2. It uses a second memory address space called control panel memory to control a video display and keyboard. Special interrupts suspend the user program and transfer control to code in control panel memory. This operation is transparent to the user and the user program.

2.2.1.2 **Control Panel Memory** -- Control panel memory is a second 32K word memory address space outside the VT278's normal 32K word range. The added memory space allows rapid context switching. The 32K words of control panel memory address space is used for the following functions:

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

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

2.2.1.4 **CRT Control** -- The display portion of the VT278 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 in any combination can be assigned to each character on an individual basis. These and other 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 Chapter 5.

2.2.1.5 **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 Chapter 5.

2.2.1.6 **RX278 Diskette Subsystem Control** -- A single 37-pin connector at the rear of the VT278 connects one or two dual RX02 diskette drives. The instruction set for the diskette control is discussed in Chapter 5.

2.2.1.7 **Printer Port** -- A 25-pin connector at the rear of the VT278 connects to a variety of EIA serial printers. The baud rate is set to 9600 when power is turned on. It may be changed to a speed of 50 to 19200 baud under program control. The instruction set is given in Chapter 5.

2.2.2 Keyboard

The keyboard connects to the VT278 CPU board via a 3-wire cable. The cable supplies +12 Vdc and ground to power the keyboard logic. The remaining wire is a bidirectional serial communication line. It carries command information (requests for keyboard data, LED information, and bell commands) from the VT278 CPU board to the keyboard. It also carries a unique address from the keyboard to the VT278 CPU board for each key pressed.

2.2.3 Power Supply

The power supply accepts either 120 or 220/240 Vac input. Voltage input is selected with a slide switch on the VT278 rear panel. A second switch indicates whether the input frequency is 50 or 60 Hz. The power supply generates +5 Vdc and $\pm$12 Vdc for use by the VT278 terminal logic.

2.2.4 Video Monitor

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

2.3 OPTION OVERVIEW

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

2.3.1 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 determined by the modem or the VT278 to a maximum of 9600 baud. The DP278 also contains firmware to simulate a simple I/O terminal with full modem control. Two versions of the DP278 are available:

1. DP278-A provides two programmable communication lines with full modem control;

2. DP278-B is similiar to DP278-A with additional circuitry to support bit synchronous (BITSTUFF) operation.

Run-time characteristics for each port may be chosen 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: 50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600, 19200 baud

The instruction set for the DP278 is given in Chapter 6.

2.3.2 RL278 Disk Subsystem Option

The RL278 disk controller controls up to four RL02 disk drives. The RL278 offers fully programmable DMA addressing. Up to 4096 words (programmable for 8 bits or 12-bits) may be transferred at one time. The instruction set for the RL278 is given in Chapter 7.

2.3.3 Extended Language Option

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

CHAPTER 3
OPERATING PROCEDURES

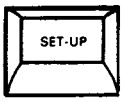
3.1 GENERAL

This chapter describes the VT278 operator controls and indicators and a number of procedures for functions such as self-test, SET-UP, and system initialization.

3.2 CONTROLS AND INDICATORS

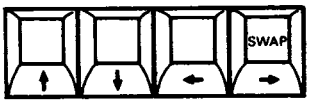
This section describes the controls and indicators on the VT278 keyboard (Figure 3-1) and on the VT278 rear panel (Figure 3-2).

3.2.1 Keyboard Controls



SET-UP

This key is used to enter and exit SET-UP mode. Paragraph 3.4 describes SET-UP mode and its options.



Cursor Control Keys

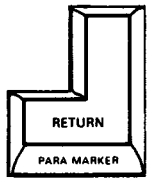
In local mode, each of these keys generates an ESCape code sequence that moves the cursor in the direction shown. In SET-UP mode, the ↑ and ↓ keys are used to increase and decrease screen brightness.

Backspace

This key transmits a backspace code (010_8)

Delete

This key transmits a delete code (177_8).



Return

This key transmits a carriage return (CR) code (015_8).

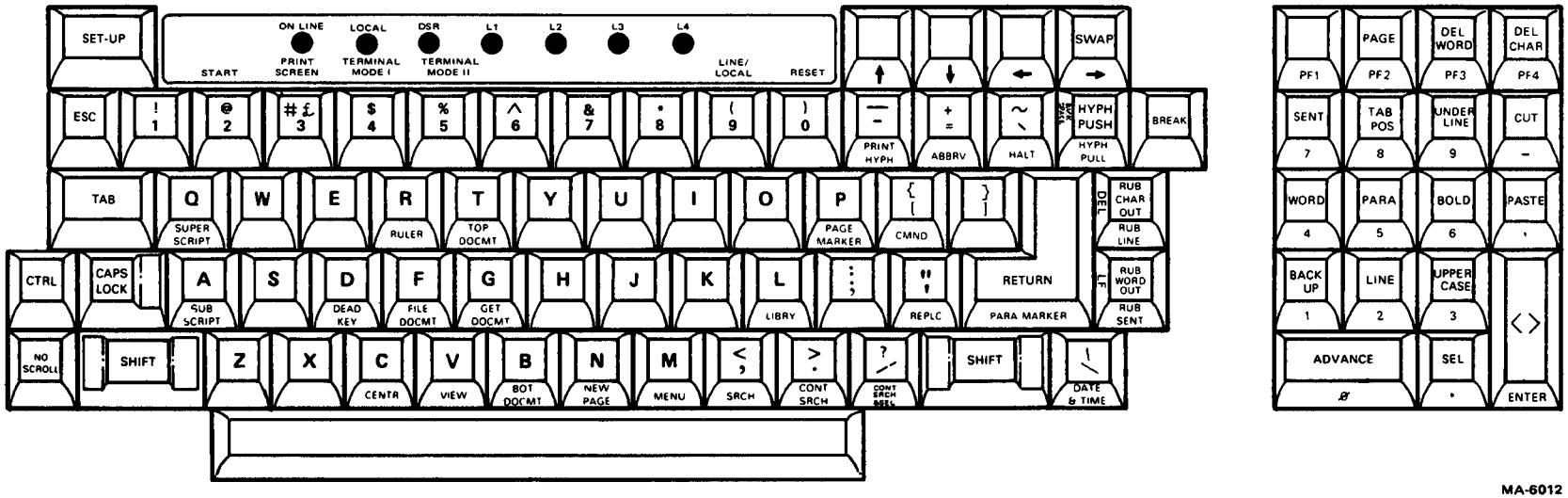


Figure 3-1 VT278 Keyboard

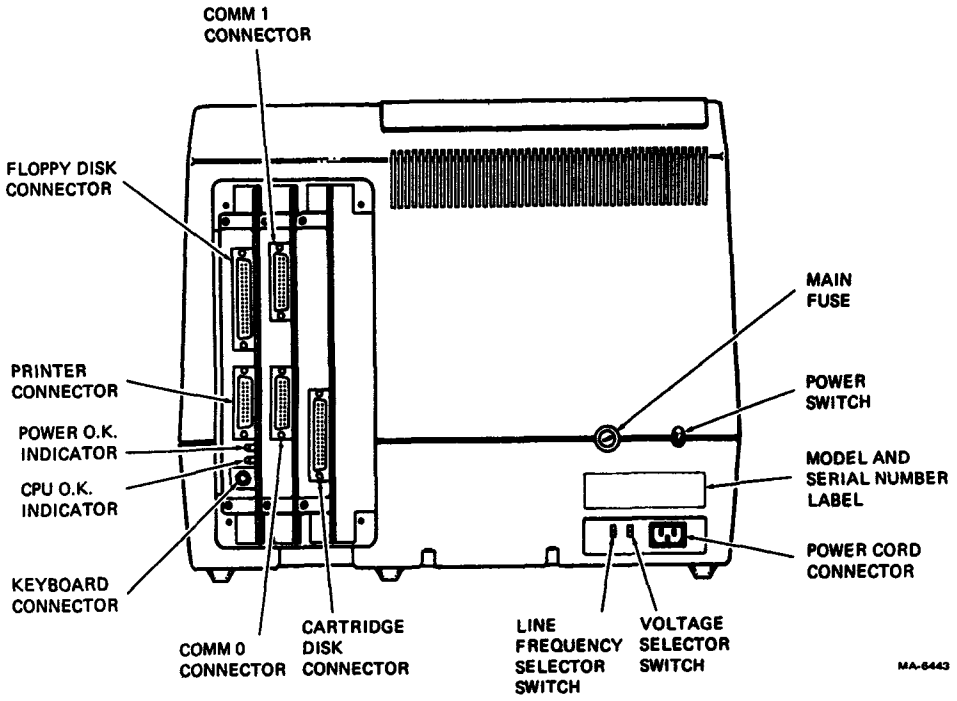
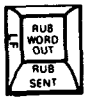
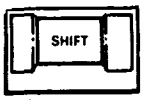


Figure 3-2 VT278 Rear Panel With I/O Cover Removed

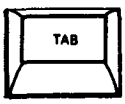


Linefeed
This key transmits a linefeed (LF) code (012_8).



Shift
When pressed, this key enables the uppercase function of all keys. If a key does not have an uppercase function the SHIFT key is disregarded.

Control
This key when pressed in combination with another key, generates a control character code.

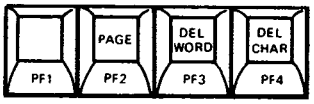


Tab
This key transmits a tab code (011_8).

ESCape
This key transmits an ESCape code (033_8).

3.2.2 Numeric Keypad

The numeric keypad enters numbers in calculator fashion. Each key generates the same character as the corresponding numeric key on the main keyboard. The **ENTER** key corresponds to the **RETURN** key. These keys may also be used as special function keys when the numeric keypad is set to application mode.



PF1--PF4 Programmed Function Keys

Each of these programmed function keys produces an ESCape code sequence that can have special meaning in a user program.

3.2.3 Keyboard LED Indicators

DSR and L1--L4

These five LEDs can each be turned on and off under program control by using the ESCape code sequences described in Chapter 5.

ON LINE

This LED is controlled by VT278 firmware and is turned on after the successful completion of the VT278 self-test. The LED is off if any of the fatal self-tests fail or when the VT278 is in local mode.

LOCAL

This LED is controlled by VT278 firmware and is only turned on when the VT278 is in LOCAL mode. To enter LOCAL mode, you must first be in SET-UP mode, as described in Paragraph 3.4.

3.2.4 Rear Panel Controls and LED Indicators

POWER switch

Push this two-position switch UP to turn power on and DOWN to turn power off.

PWR OK

This LED lights if the VT278 passes its power supply tests during self-test. The LED does not light if the power supply is not operating correctly.

CPU OK

This LED blinks once per second if the VT278 completes a very basic set of tests. If an error in any of these tests occurs, the LED may be ON or OFF. Either continuous condition indicates a basic failure.

3.3 SELF-TEST

A self-test mode in the VT278 automatically tests the condition of the VT278 when power is turned on. To turn power on, push the POWER switch on the VT278 rear panel to the UP position (Figure 3-2).

After the basic tests, there are two categories of self-test errors: fatal, and nonfatal. Fatal errors indicate that the VT278 is inoperable because of an internal failure in, for example, memory or the power supply. Fatal errors prevent operation of the VT278.

Nonfatal errors indicate that the printer port, real-time clock, or interrupt system are malfunctioning. The VT278 is still operational even though a nonfatal error condition exists. However, any user program that attempts to use the feature that caused the nonfatal error may produce unpredicatable results.

You can determine the success or failure of VT278 self-tests by watching the CPU OK and PWR OK LEDs on the VT278 rear panel (Figure 3-2), the keyboard LEDs in Figure 3-1, and the display.

3.3.1 Power Tests

When power is applied to the VT278 (POWER switch up), the power supply first tests to see that all its output voltages are within tolerance. If they are, the PWR OK LED on the rear panel lights. If PWR OK does not light, the remaining tests will not be performed.

3.3.2 CPU OK Tests

The following three tests are performed after the power test. If any one of the tests fails, CPU OK lights continuously, and the remaining tests will not be performed.

T1--control panel ROM memory checksum test
T2--control panel RAM memory read/write test
T3--keyboard response to scan initiation command

3.3.3 General Tests

The tests in this group are summarized in Table 3-1. The success or failure of each test can be determined by watching the CPU OK LED and keyboard LEDs. Table 3-1 shows the status of the LEDs in the event of a test failure. Tests T4 through T9 produce fatal errors; tests T1Ø through T12 produce nonfatal errors.

If the VT278 passes all tests, CPU OK blinks, the word **SET-UP** appears in blinking bold characters at the bottom of the display, and only ON LINE on the keyboard lights.

If any of the basic tests in Table 3-1 fails, CPU OK lights continuously.

If all fatal error tests pass, but one of the nonfatal errors occurs, CPU OK blinks, the word **SET-UP** appears at the bottom of the display, and the LEDs determine the failed test.

In Table 3-1, "X" means the LED is on, "O" means the LED is off. Notice there are seven LEDs as shown in Figure 3-1.

Table 3-1 General Self-Test Summary

Test No.	Description	CPU OK LED	LEDs	Screen Display
Fatal Error Tests				
T4	Display buffer memory	On	O X X X X X X	None
T5	CRT controller	On	O X X X X X O	None
T6	Main memory read/write	On	O X X X X O X	None
T7	Second main memory test	On	O X X X X O O	None
T8	HLT instruction or	On	O X X X O X X	None
T9	Panel request failure	On	O X X X O X O	None
Nonfatal Error Tests				
T1Ø	Printer port loopback	Blinks	O X X X O O X	SET-UP
T11	Real-time clock	Blinks	O X X X O O O	SET-UP
T12	Printer and real-time clock	Blinks	O X X O X X X	SET-UP
T13	Undefined interrupt	Blinks	O X X O X X O	SET-UP

X=LED on
O=LED off

3.4 SET-UP MODE PROCEDURES

When the self-tests described in Paragraph 3.3 are completed, the VT278 is in SET-UP mode and the word **SET-UP** appears at the bottom of the display screen. You can enter and exit SET-UP mode during normal system operation, as described in Paragraph 3.5. The following paragraphs describe the functions you can perform in SET-UP mode.

3.4.1 Change Screen Brightness

You can change the display screen brightness as follows:

1. To increase brightness press ↑ (cursor up).

2. To decrease brightness press ↓ (cursor down).

Both keys auto-repeat and continue to step the brightness toward maximum or minimum intensity if you hold the key down. There are a total of 16 intensity levels (Ø to maximum). When power is turned on to the VT278, the intensity is set at mid-point.

3.4.2 Repeat Self-Tests

You can repeat the self-tests described in Paragraph 3.3 by pressing the Ø key while in SET-UP mode. The VT278 returns to SET-UP mode when the tests are completed.

3.4.3 LOCAL Mode

LOCAL mode allows you to fully test the keyboard and display by typing characters and ESCape code sequences on the keyboard and verifying the results on the display screen. If you enter SET-UP mode and LOCAL mode during normal system operation, (Paragraph 3.5), the characters you type on the keyboard are sent directly to the screen. They are not sent to any user program that might be running in main memory.

To enter LOCAL mode while in SET-UP mode:

1. Press the 9 key.
   The ON LINE LED turns off and the LOCAL LED turns on.

2. Press the **SET-UP** key.
   The word **SET-UP** disappears from the display screen, the cursor moves to the home position, and you can begin typing on the keyboard.

To exit LOCAL mode and return to SET-UP mode:

1. Hold down the **CTRL** key, press the **SET-UP** key; then release both keys.
   The word SET-UP reappears at the bottom of the display screen.

2. Press the 9 key.
   The LOCAL LED turns off and the ON LINE LED turns on.

3.4.4 RX278 Bootstrap

If the VT278 system has an RX278 diskette subsystem, you can bootstrap an operating system (e.g., COS-31Ø) from floppy disk drive Ø by pressing the 2 key while in SET-UP mode. The word **SET-UP** disappears from the screen, the cursor moves to the home position, and a message from the operating system appears on the screen. At this point the VT278 is no longer in SET-UP mode but is under control of the operating system.

3.4.5 RL278 Bootstrap

If the VT278 system has an RL278 disk subsystem, you can bootstrap an operating system (e.g., COS-31Ø) from hard disk drive Ø by pressing the 6 key while in SET-UP mode. The word **SET-UP** disappears from the screen, the cursor moves to the home position, and a message from the operating system appears on the screen. At this point the VT278 is no longer in SET-UP mode but is under control of the operating system.

3.4.6 Print Screen

When the VT278 is in SET-UP mode, you can print the contents of the display screen by pressing the 3 key. A printer must be

connected to the serial printer port. The printer port is initially set to 9600 baud during self-test, but the baud rate can be changed under program control (see Chapter 5).

3.4.7 Terminal Mode Communication

The VT278 has two terminal mode operations when equipped with the DP278 communication option:

1. Terminal mode I requires a hardwired connection between the DP278 and a host computer. In this mode, the VT278 operates similar to a VT100 video display terminal. Connections must be made from communication port 0 and involve the following signals.

Signal Name	Connector Pin	EIA RS232C	EIA RS449	CCITT V24	DIN 66020
Transmitted data	2	BA	SD	103	D1
Received data	3	BB	RD	104	D2
Signal ground	7	AB	SG	102	E2
Protective ground*	1	AA	--	101	E1

* Optional connection

2. Terminal mode II is similar to mode I, but the connection between the DP278 and the host computer must be made via a modem. The modem must be connected to communication port 0. Configuration information for matching the DP278 to a selected modem is included in the DECmate Pocket Service Guide.

Both modes permit a program to be down-line loaded from the host computer to the VT278. Programming details for down-line loading are provided in Chapter 6.

3.4.7.1 Initiating Terminal Mode I from SET-UP Mode -- The DP278 must be installed in the VT278 and hardwired to the host computer. Pressing the 4 key causes the word SET-UP to disappear from the screen and puts the VT278 into "simple terminal mode." In this mode any key pressed on the keyboard is transmitted via communication port 0, and any character received as input from port 0 appears on the screen.

When you press the 4 key, the following occurs:

1. The DP278 enters Parameter SET-UP mode. A parameter screen is displayed which allows you to change the baud rate and/or the communication mode. The parameters may be changed by typing a number from Table 3-2 for the baud rate and a letter from Table 3-3 for the communication mode.

2. To exit Parameter SET-UP mode, press the RETURN key.

Table 3-2 Baud Rate Selection

Rate	Press Key
110*	1
300	2
600	3
1200	4
2400	5
4800	6

* Communication baud rate defaults to 300 when alternate baud rate not specified.

3. The DP278 responds by displaying the word **CONNECTED** in the upperleft corner of the screen.

4. The VT278 will now function as a terminal connected to the host computer.

3.4.7.2 Initiating Terminal Mode II from SET-UP Mode -- Pressing the 5 key with the DP278 installed causes the word **SET-UP** to disappear from the screen and allows the VT278 to perform similar to terminal mode I. The main difference is that in terminal mode II the DP278 provides the modem control signals required for communicating through a modem.

When you press the 5 key, the following occurs:

1. The DP278 enters Parameter SET-UP mode. A parameter screen is displayed which allows you to change the baud rate and/or the communication mode. You can change the parameters by typing a number from Table 3-2 for the baud rate and a letter from Table 3-3 for the communication mode.

2. To exit Parameter SET-UP mode, press the **RETURN** key.

3. The DP278 responds by displaying the word **CALL** in the upperleft corner of the screen. This reminds you to dial the host computer and make the telephone connection via the modem.

NOTE: You must press the **RETURN** key in response to the **CALL** message. This causes the DP278 to try to connect to the host. After you press **RETURN**, you have 30 seconds to complete the call.

4. The DP278 attempts to complete the connection. When it does, the word **CONNECTED** appears below the word **CALL**.

5. The VT278 now functions as a terminal connected to the host computer.

Table 3-3 Communication Mode Selection

Mode	Press Key
No parity, 1 stop bit, 8 data bits, 80 display columns*	A
No parity, 1 stop bit, 8 data bits, 132 display columns	B
No parity, 2 stop bits, 8 data bits, 80 display columns	C
No parity, 2 stop bits, 8 data bits, 132 display columns	D
Odd parity, 1 stop bit, 7 data bits, 80 display columns	E
Odd parity, 1 stop bit, 7 data bits, 132 display columns	F
Odd parity, 2 stop bits, 7 data bits, 80 display columns	G
Odd parity, 2 stop bits, 7 data bits, 132 display columns	H
Even parity, 1 stop bit, 7 data bits, 80 display columns	I
Even parity, 1 stop bit, 7 data bits, 132 display columns	J
Even parity, 2 stop bits, 7 data bits, 80 display columns	K
Even parity, 2 stop bits, 7 data bits, 132 display columns	L

* Communications mode defaults to no parity, 1 stop bit, 8 data bits, 80 display columns (letter A) when alternate mode not specified.

3.4.7.3 Changing Parameters -- You may want to return to Parameter SET-UP mode to make changes after you are connected to the host computer.

1. Type **CTRL-K, RETURN**

2. Make the desired changes per Tables 3-2 and 3-3

3. Press **RETURN.**

The VT278 returns to the point it was at before you typed **CTRL-K.**

3.4.7.4 Stopping Communication -- To disconnect the communication link:

1. Type **CTRL-K**

2. Press the **D** key.

The communication line is disconnected and you are back in Parameter SET-UP mode.

3.5 ENTERING AND EXITING SET-UP MODE

You can enter SET-UP mode from the keyboard when the VT278 system is under control of a user program. Then, any of the SET-UP procedures in Paragraph 3.4 can be performed. You can also enter SET-UP mode directly from a user program by executing a PR2 memory control instruction as described in Chapter 4.

3.5.1 Entering SET-UP Mode

To enter SET-UP mode from the keyboard:

1. Hold down the **CTRL** key, press the **SET-UP** key; then release both keys.

2. The bottom row of the display is saved, and then the bottom row is cleared.

3. The word **SET-UP** appears at the left of the bottom row in bold, blinking characters.

4. Any of the SET-UP functions in Paragraph 3.4 can now be performed.

3.5.2 Exiting SET-UP Mode

To exit SET-UP mode:

1. Press the **SET-UP** key.

2. The bottom line of the display that was saved and erased is restored to the screen.

3. The VT278 returns control to the user program that was interrupted when SET-UP mode was entered.

NOTE: Indiscriminate use of this feature may cause certain user programs to lose data from a serial line since normal interrupts are inhibited during SET-UP mode.

CHAPTER 4
PROGRAMMING THE VT278

4.1 CENTRAL PROCESSOR UNIT

The VT278 central processor unit (6120 CPU) is comprised of 12 special purpose registers, control logic, and an arithmetic logical unit (ALU). The ALU performs both arithmetic and logical operations such as AND, TAD, and logical OR. The following paragraphs describe each special purpose register in the CPU.

4.1.1 Accumulator

The accumulator (AC) is a 12-bit register that performs arithmetic and logical operations. Data words may be fetched from memory to the AC or stored from the AC into memory. Arithmetic and logical operations involve two operands, one held in the AC and the other fetched from memory. The result of the operation is left in the AC. The AC may be cleared, complemented, tested, incremented, or rotated under program control. The AC also serves as the input-output register for programmed I/O.

4.1.2 Link

The link (L) is a 1-bit register that serves as a high-order extension of the AC. It is used as a carry register for 2's complement arithmetic. A carry out of the ALU complements the link. The link may be set, cleared, or complemented under program control. It may also be rotated left or right as part of the AC.

4.1.3 Multiplier Quotient Register

The multiplier quotient (MQ) is a program accessible 12-bit temporary register. The contents of the AC may be transferred to the MQ for temporary storage. The MQ can be ORed with the AC and the result stored in the AC. The contents of the AC and the MQ may also be exchanged.

4.1.4 Program Counter

The 12-bit program counter (PC) contains the address of the memory location from which the next instruction is normally fetched. During an instruction fetch, the PC is incremented by 1. When there is a branch to another address in memory, the branch address is set into the PC. Branching normally takes place under program control. A skip (SKP, SMA, SZA, SNL) instruction further increments the PC by 1, thus causing the next instruction to be skipped. The skip instruction may be unconditional or conditional on the state of the AC and/or link. During an input-output operation, a device can also cause the next instruction to be skipped.

4.1.5 Instruction Register

During an instruction fetch, the 12-bit instruction register (IR) contains the instruction to be executed by the 6120.

4.1.6 Stack Pointers

The stack pointers (SP1 and SP2) are two 12-bit registers which hold the address of the next stack storage location. PPCX or PACX instructions cause post-decrement of the contents of stack pointer SPX. RTNX or POPX cause a preincrement of the contents of the stack pointer. Stack pointers are loaded from, and read into, the AC. They may also be used as program-controlled temporary registers.

4.1.7 Instruction Field

The 3-bit instruction field (IF) holds the memory field from which all instructions, all indirect address pointers, and all directly addressed operands are obtained. It may be read into the AC, and loaded from the instruction buffer (IB). It is cleared by RESET.

4.1.8 Instruction Buffer

The 3-bit instruction buffer (IB) serves as a holding register for instructions which change the IF. Instead of changing the IF directly, field bits are loaded into the IB, and transferred to the IF at the next JMP, JMS, RTN1, or RTN2. The IB may be loaded from instruction bits, from the AC, or from the ISF. The IB is cleared by RESET.

4.1.9 Data Field

The 3-bit data field (DF) holds the memory field from which all indirectly addressed operands are obtained. The DF may be loaded from instruction bits, from the AC, or from the DSF. It may be read into the AC. It is cleared by RESET.

4.1.10 Instruction Save Field

The 3-bit instruction save field (ISF) is loaded with the contents of the IF upon granting an interrupt. The ISF may be read into the AC. It is cleared by RESET.

4.1.11 Data Save Field

The 3-bit data save field (DSF) is loaded with the contents of the DF upon granting an interrupt. The DSF may be read into the AC. It is cleared by RESET.

4.1.12 GT Flag

The GT flag is a 1-bit register that can be loaded or read under program control.

4.2 MULTIPLEX BUSES

There are three buses internal to the VT278 that carry data and/or instructions. The buses and their functions are discussed in the following paragraphs.

4.2.1 DX Bus

The 12-bit input/output multiplexer bus handles data, address and instruction transfers between the 6120 and the main memory, panel memory, and peripheral devices on a time-multiplexed basis.

4.2.2 CØ and C1 Bus

The CØ and C1 bus is a bidirectional time-multiplexed bus that transmits the two MSBs of the extended memory address (instruction or data field), and the status of the link (L) and GT flags. This bus also receives I/O device transfer control commands.

4.2.3 EMA2 Bus

EMA2 is an output bus from the 6120 that transmits the LSB of the extended memory address and the status of the interrupt enable flip-flop (IEFF) on a time-multiplexed basis.

4.3 MEMORY

There are three general classes of instructions:

Memory reference instructions (MRI)
Operate instructions (OPR) and
Input/output instructions (IOT).

IOT instructions are further subdivided into external I/O instructions and internal control instructions.

4.3.1 General Description

Instructions are 12-bit words stored in memory. The 6120 makes no distinction between instructions and data. It can manipulate instructions as stored variables or execute data as instructions when programmed to do so.

Before proceeding further, we will discuss the memory organization with which the 6120 interfaces.

4.3.2 Memory Organization

The 6120 has an addressing capacity of 32,768 12-bit words. The memory system is organized into 4096 word groups, called memory fields, as shown in Figure 4-1. The first 4096 words of memory are field Ø. If a full 32K block of memory is installed, the uppermost memory field is numbered 7. Two 32K word blocks of memory may be connected to the 6120. One of these blocks is main memory and the other is panel memory. In any given memory field, every location has a unique 4-digit octal (12-bit binary) address, ØØØØ--7777 (ØØØØ--4095 decimal).

Each memory field is subdivided into 32 pages of 128 words each. Memory pages are numbered sequentially from page ØØ, containing octal addresses ØØØØ--Ø177, to page 37 (octal), containing octal addresses 76ØØ--7777. The most significant 5 bits of a 12-bit memory address denote the page number; the 7 low order bits specify the address of the memory location within the given page.

During an instruction fetch cycle, the 6120 fetches the instruction pointed to by the instruction field and the PC. The PC is incremented by 1. The PC now contains the address of the 'next' sequential instruction.

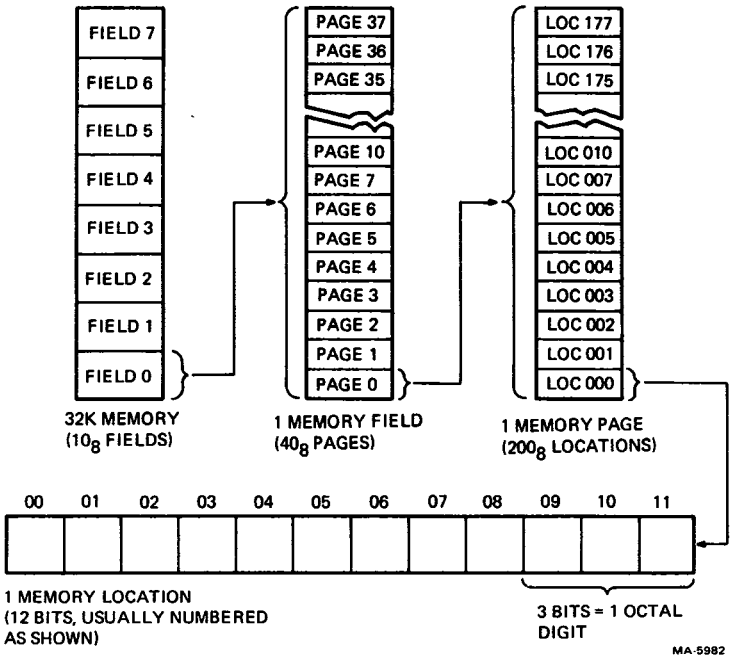


Figure 4-1 VT278 Main Memory Organization

4.4 TWO'S COMPLEMENT NOTATION

Data is represented in two's complement integer notation. In this system of notation, the negative of a number is formed by complementing each bit in the data word and adding "1" to the complemented number. The sign is indicated by the most significant bit. In the 12-bit word used in the 6120, when bit 0 is a "0," it denotes a positive number and when bit 0 is a "1," it denotes a negative number. The number range for this system is +3777 to -4000 octal (+2047 to -2048 decimal).

4.5 INSTRUCTIONS

This section describes the VT278 instruction set, which includes:

- Memory reference instructions (MRI) that operate on data stored in a memory location.
- Operate instructions that allow manipulation and/or testing of data in the accumulator and link.
- External I/O instructions that transfer information between a peripheral device and main memory.
- Internal control operation instructions which are similar to external I/O instructions, but control internal facilities such as stacks.

4.5.1 Memory Reference Instructions

Memory reference instructions (MRI) operate on the contents of a memory location or use the contents of a memory location to operate on the AC or the PC. Figure 4-2 shows the MRI format. Bits Ø--2 of a memory reference instruction specify the operation code, or opcode, and the 9 low-order bits specify the operand address. Bits 5--11, the page address, identify the location of the operand on a given page, but they do not identify the page itself. The page is specified by bit 4, called the page bit. If bit 4 is a "Ø," the page address is interpreted as a location on page Ø. If bit 4 is a "1," the page address is interpreted to be on the current page. The entire 12-bit address, consisting of the 7 low-order bits from the instruction and either Ø or the current page bits in the 5 high-order bits is known as the instruction address, or IA. The instruction field provides the 3 high-order bits of the complete 15-bit address (IA).

Other locations are addressed by using bit 3. When bit 3 is a "Ø," the operand is directly addressed, and IA is the location of the operand. When bit 3 is a "1," the operand is indirectly addressed, and the contents of IA specify the location of the operand. To address a location that is not on page Ø or the current page, the absolute address of the desired location is stored in one of the 256 directly-addressable locations as a pointer address. The instruction addresses the operand indirectly through this pointer. Upon execution, the MRI operates on the contents of the location identified by the address contained in the pointer location. The pointer is obtained from the current instruction field; the data is in the current data field (DF).

It should be noted that locations ØØ1Ø--ØØ17 (octal) in page Ø of any field are autoindexed. If these locations are addressed as indirect pointers, the contents are incremented by 1 and restored before they are used as the operand address. These locations may, therefore, be used for indexing applications. During the memory write operation, the DF appears on CØ, C2, and EMA2. Indirect reference to auto index registers from page Ø work as defined whether the page bit is "1" or "Ø."

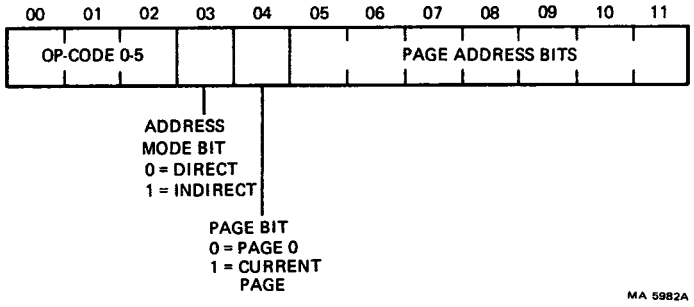


Figure 4-2 Memory Reference Instruction Format

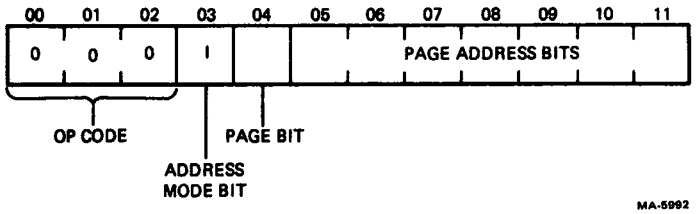
The memory reference instructions are as follows.

AND	Logical AND to AC
TAD	Two's complement add
ISZ	Increment and skip if zero
DCA	Deposit and clear the accumulator
JMS	Jump to subroutine
JMP	Jump

The following notation describes the memory reference instructions.

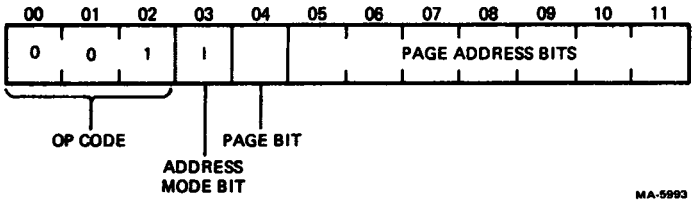
Symbol	Definition
IA	Instruction address -- a 12-bit address derived from the memory reference instruction word and the PC. Bits Ø--4 are Ø if bit 4 of the instruction was Ø. Bits Ø--4 are equal to bits Ø--4 of the PC when the instruction was fetched if bit 4 of the instruction was a 1. Bits 5--11 are bits 5--11 of the instruction.
(IA)	"The contents of the memory location pointed to by the instruction address."
EA	Effective address -- equal to IA if instruction bit 3=Ø; equal to (IA) if instruction bit 3=1.
((IA))	"The contents of the location pointed to by (IA)."
Opcode	Bits Ø--2 of the instruction
<=	"Is replaced by"
I	Bit 3 of the instruction
*	Denotes logical AND operation
V	Denotes logical OR operation
(IB)	Contents of the instruction buffer

4.5.1.1 AND



Purpose	Logical AND to AC
Execution time (us)	Direct 2.76 Indirect 3.95 Autoindex 5.13
Operation	Logical AND direct (I=Ø): (AC)<=(AC)*(IA).
	Logical AND indirect autoindex (I=1, IA is not ØØ1Ø-ØØ17): (AC)<=(AC)*((IA)).
	Logical AND indirect autoindex (I=1, IA is ØØ1Ø-ØØ17): (IA)<=(IA)+1 (AC)<=(AC)*((IA)).
Operation code	Ø
Description	The contents of the effective address (EA) is combined with the contents of the AC by a bitwise AND operation. The result is left in the AC, memory is unchanged, and the original content of the AC is lost.

4.5.1.2 TAD



Purpose	2's complement add
Execution time (us)	Direct 2.76 Indirect 3.95 Autoindex 5.13
Operation	Binary add direct (I=Ø): (AC)<=(AC)+(IA).

Binary add indirect (I=1, IA is not 0010-0017):
(AC<=(AC)+((IA)).

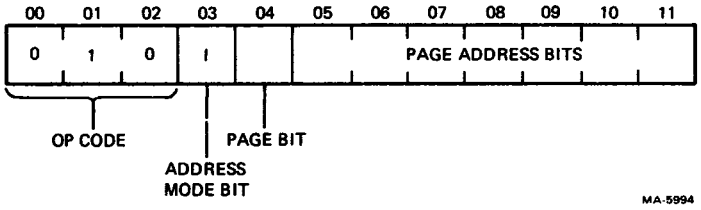
Binary add indirect autoindex (I=1, IA is 0010-0017):
(IA)<=(IA)+1
(AC)<=(AC)+((IA)).

Operation code: 1

Description: The contents of the effective address (EA) are added with the contents of the AC and the result is stored in the AC. Memory is unchanged and the original content of the AC is lost. If there is a high-order carry from AC0, the link is complemented.

If the AC is initially cleared, this instruction acts as a load from memory.

4.5.1.3 ISZ



Purpose: Increment and skip if zero

Execution time (us)	Direct	Indirect	Autoindex
	3.55	4.74	5.92 (no skip occurs)
	4.34	5.52	6.71 (skip occurs)

Operation

Increment and skip if zero direct (I=0):
(IA)<=(IA)+1
If (IA)=0 then (PC)<=(PC)+1.

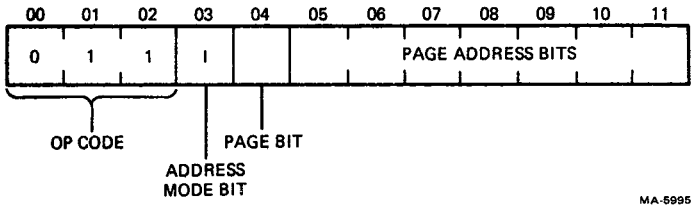
Increment and skip if zero indirect (I=1, IA is not 0010-0017):
((IA))<=((IA))+1
If ((IA))=0 then (PC)<=(PC)+1.

Increment and skip if zero autoindex (I=1, IA is 0010-0017):
(IA)<=(IA)+1
((IA))<=((IA))+1
If ((IA))=0 then (PC)<=(PC)+1.

Operation code 2

Description The content of the effective address (EA) is incremented by 1 and restored to memory. If the content of the EA becomes Ø, the PC is incremented by 1 to skip the next sequential instruction. If the content of the EA does not become Ø, the next instruction is executed.

4.5.1.4 DCA



Purpose Deposit and clear the accumulator.

Execution time (us)	Direct	Indirect	Autoindex
	2.76	3.95	5.13

Operation

Deposit and clear the accumulator direct (I=Ø):
(IA)<=(AC)
(AC)<=Ø.

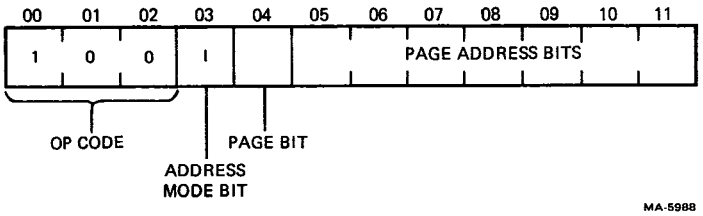
Deposit and clear the accumulator indirect (I=1, IA is not ØØ1Ø-ØØ17):
((IA))<=(AC)
(AC)<=Ø.

Deposit and clear the accumulator autoindex (I=1, IA is ØØ1Ø-ØØ17):
(IA)<=(IA)+1
((IA))<=(AC)
(AC)<=Ø.

Operation code 3

Description The content of the AC is stored in the effective address (EA) and the AC is set to zero. The original content of the effective address is lost.

4.5.1.5 JMS



Purpose — Jump to subroutine

Execution time (us)	Direct	Indirect	Autoindex
	2.76	3.95	5.13

Operation

Jump to subroutine direct (I=Ø):
(IF)<=(IB)
(IA)<=(PC)
(PC)<=IA+1.

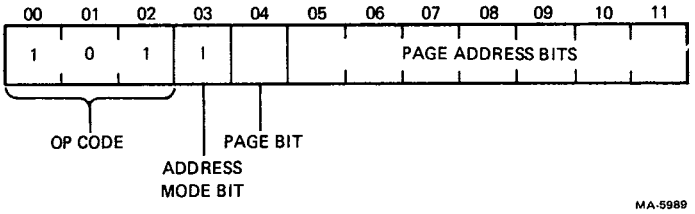
Jump to subroutine indirect (I=1, IA is not ØØ1Ø-ØØ17):
(IF)<=(IB)
((IA))<=(PC)
(PC)<=(IA)+1.

Jump to subroutine autoindex (I=1, IA is ØØ1Ø-ØØ17):
(IA)<=(IA)+1
(IF)<=(IB)
((IA))<=(PC)
(PC)<=(IA)+1.

Operation code — 4

Description — The content of the PC is stored in the EA. The PC is then loaded with 1 more than the EA, so that the instruction that follows the EA is the next instruction to be executed. Therefore, the EA contains the return address from the subroutine. The content of the AC is not affected.

4.5.1.6 JMP



Purpose	Jump

Execution time (us)	Direct	Indirect	Autoindex
	1.58	2.76	3.95

Operation

Jump direct (I=Ø):
(IF)<=(IB)
(PC)<=IA.

Jump indirect (I=1, IA is not ØØ1Ø-ØØ17):
(IF)<=(IB)
(PC)<=(IA).

Jump autoindex (I=1, IA is ØØ1Ø-ØØ17):
(IA)<=(IA)+1
(IF)<=(IB)
(PC)<=(IA).

Operation code: 5

Description: The EA is loaded into the PC so that the instruction stored at this address is the next instruction to be executed. The original content of the PC is lost. The content of the AC is not affected.

4.5.2 Operate Instructions

Operate instructions have an opcode of 7 (111) and are divided into 3 groups. Group 1 instructions have a 0 in bit 3. They perform logical operations on the contents of the AC and link. Group 2 instructions have a 1 in bit 3 and a 0 in bit 11. They primarily test the contents of the AC and then conditionally skip the next instruction. Group 3 instructions have a 1 in bit 3 and a 1 in bit 11. They perform logical operations on the contents of the AC and MQ. The basic operate instruction format is shown in Figure 4-3.

In most cases, operate instructions from a group may be microprogrammed with other operate instructions of the same group. In these instances, the actual code for a microprogrammed combination of two or more operate instructions is the bit-wise logical OR of the octal codes for the individual operate instructions. When more than one operation is microprogrammed into a single instruction, the operations are performed in a prescribed sequence, with logical sequence number 1 performed first, number 2 performed second, number 3 performed third, and so on. Two operations with the same logical sequence number within a given group are performed simultaneously.

4.5.2.1 Group 1 Operate Instructions -- Figure 4-4 shows the instruction format of group 1 operate instructions. Any one of bits 4--11 may be set (loaded with a binary 1) to indicate a specific group 1 operate instruction. If more than one of these

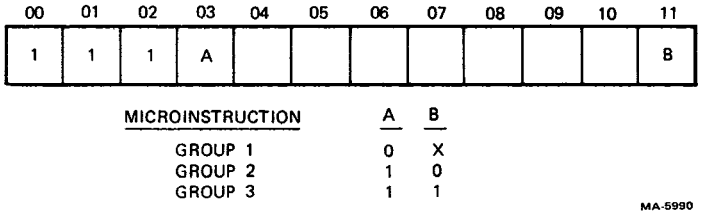


Figure 4-3 Basic Operate Instruction Format

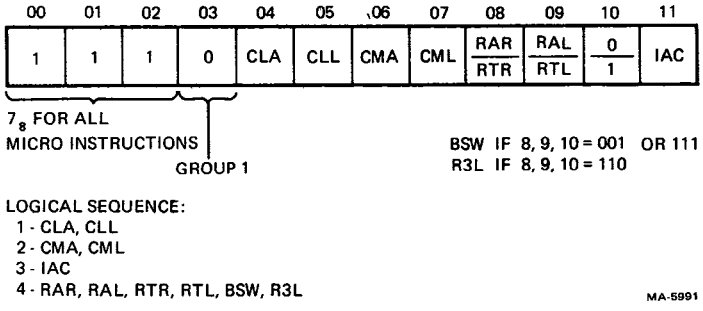


Figure 4-4 Group 1 Microinstruction Format

bits is set, the instruction is a microprogrammed combination of group 1 instructions, which will be executed according to the logical sequence shown in Figure 4-4. Bits 8, 9, and 10 are an exception to this rule; these bits specify 1 of 8 possible types of rotate instructions.

Table 4-1 lists commonly used group 1 instructions, their assigned mnemonics, octal number, instruction format, logical sequence, and the operation they perform. All group 1 microinstructions require 2.37 microseconds, but RTR, RTL, and BSW require 3.16 microseconds.

Table 4-2 lists commonly used microprogrammed group 1 microinstructions.

It is often useful to generate a constant by microprogramming group 1 microinstructions that manipulate the AC and the link. By microprogramming a constant, you can save using a memory location. Table 4-3 contains a number of instructions that produce some commonly used constants.

Table 4-1 Group 1 Operate Instructions

Mnemonic	Octal	Sequence	Description
NOP	7000	1	No operation -- Causes a 1-cycle delay in program execution, without affecting the state of the computer. May be used for timing synchronization or as a convenient means of deleting another instruction from a program.
IAC	7001	3	Increment accumulator -- The content of the accumulator is incremented by 1. A carry out of the accumulator complements the link.
BSW	7002	4	Byte swap -- The content of the six low-order bits of the AC is exchanged with the content of the six high-order bits. That is, bits AC<0:5> are exchanged with bits AC<6:11>. The content of the link is not affected.
RAL	7004	4	Rotate accumulator left -- The content of AC<1:11> is shifted into AC<0:10>. The content of AC0 is shifted into the link, and the content of the link is shifted into AC11.
RTL	7006	4	Rotate two left -- Equivalent to two consecutive RAL operations.

Table 4-1 Group 1 Operate Instructions (Cont)

Mnemonic	Octal	Sequence	Description
RAR	7010	4	**Rotate accumulator right** -- The content of AC<0:10> is shifted into AC<1:11>. The content of the link is shifted into AC0, and the content of AC11 is shifted into the link.
RTR	7012	4	**Rotate two right** -- Equivalent to two consecutive RAR operations.
R3L	7014	4	**Rotate three left** -- Rotate the AC (but not the link) three places to the left. AC0 is rotated into AC9, AC1 is rotated into AC10.
CML	7020	2	**Complement link** -- The content of the link is complemented.
CMA	7040	2	**Complement accumulator** -- The content of each bit of the AC is complement. This has the effect of replacing the content of the AC with its 1's complement.
CIA	7041	2,3	**Complement and increment accumulator** -- The content of the AC is replaced with its 2's complement. This is a microprogrammed combination of CMA and IAC.
CLL	7100	1	**Clear link** -- The link is loaded with a binary 0.
STL	7120	1,2	**Set the link** -- The link is loaded with a binary 1. This is a microprogrammed combination of CLL and CML.
CLA	7200	1	**Clear accumulator** -- Each bit of the AC is loaded with a binary 0.
GLK	7204	1,4	**Get the link** -- The AC is cleared and the content of the link is shifted into AC11 while a 0 is shifted into the link. This is a microprogrammed combination of CLA and RAL.
STA	7240	1,2	**Set the accumulator** -- Each bit of the AC is loaded with a binary 1. This is a microprogrammed combination of CLA and CMA. The AC will contain 7777.

Table 4-2 Microprogrammed Combinations of Group 1 Microinstructions

Mnemonic	Octal	Sequence	Description
CLL RAL	7104	1,4	Clear the link and rotate AC and link left.
CLL RTL	7106	1,4	Clear the link and rotate AC and link two left.
CLL RAR	7110	1,4	Clear the link and rotate AC and link right.
CLL RTR	7112	1,4	Clear the link and rotate AC and link two right.
CLA IAC	7201	1,3	Clear and increment the AC. The AC will contain 0001.
CLA CLL	7300	1	Clear the AC and link.

Table 4-3 Constants Generated By Microprogrammed Group 1 Microinstructions

Constant		Instruction	Microprogrammed
Octal	Decimal	(Octal)	Combination
0001	1	7301	CLA CLL IAC
0002	2	7305	CLA CLL IAC RAL
0002	2	7326	CLA CLL CML RTL
0003	3	7325	CLA CLL CML IAC RAL
0004	4	7307	CLA CLL IAC RTL
0006	6	7327	CLA CLL CML IAC RTL
0010	8	7315	CLA CLL IAC R3L
0100	64	7303	CLA CLL IAC BSW
2000	1024	7332	CLA CLL CML RTR
3777	2047	7350	CLA CLL CMA RAR
4000	-2048	7330	CLA CLL CML RAR
5777	-1025	7352	CLA CLL CMA RTR
6000	-1024	7333	CLA CLL CML IAC RTR
7775	-3	7346	CLA CLL CMA RTL
7776	-2	7344	CLA CLL CMA RAL
7777	-1	7340	CLA CLL CMA

4.5.2.2 Group 2 Operate Instructions -- Figure 4-5 shows the instruction format of group 2 operate instructions. Bits 4--10 may be set to indicate a specific group 2 instruction. If more than one of bits 4--7, 9, or 10 is set, the instruction is a microprogrammed combination of instructions. It will be executed according to the logical sequence shown in Figure 4-5.

Skip instructions may be microprogrammed with CLA, OSR, or HLT instructions. Skip instructions with a 0 in bit 8, however, may not be microprogrammed with skip instructions with a 1 in bit 8. When two or more skip instructions are microprogrammed into a single instruction, the resulting condition on which the decision will be based is the logical OR of the individual conditions when bit 8 is 0; or, when bit 8 is 1, the decision will be based on the logical AND.

Table 4-4 describes each of the group 2 microinstructions. Table 4-5 lists the legal combinations of skip and CLA instructions. Logical combinations of skip and OSR and HLT instructions are also allowed.

All group 2 microinstructions require 2.76 microseconds, but OSR requires 3.16 microseconds.

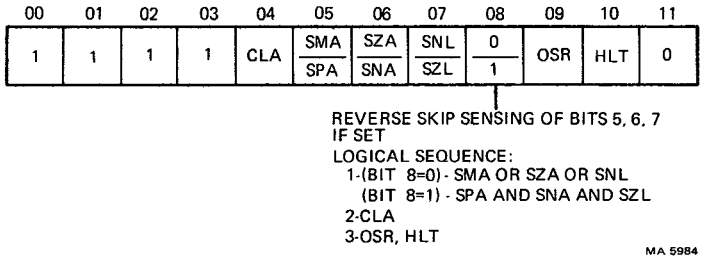


Figure 4-5 Group 2 Microinstruction Format

Table 4-4 Group 2 Operate Instructions

Mnemonic	Octal	Sequence	Description
HLT	7402	3	**Halt** -- Stops execution of the program in main memory.
OSR	7404	3	**Logical OR with status bits** -- The status bits (Paragraph 4.5.2.3) are inclusively ORed with the AC. The result is stored in the AC. The content of the status bits is not affected.
SKP	7410	1	**Skip** -- The content of the PC is incremented by 1, to skip the next sequential instruction.

Table 4-4 Group 2 Operate Instructions (Cont)

Mnemonic	Octal	Sequence	Description
SNL	7420	1	Skip on non-zero link -- The content of the link is sampled. If the link contains a 1, the content of the PC is incremented to skip the next sequential instruction. If the link contains a 0, the next instruction is executed.
SZA	7440	1	Skip on zero accumulator -- The content of each bit of the AC is sampled. If every bit contains a 0, the content of the PC is incremented to skip the next sequential instruction. If any bit contains a 1, the next instruction is executed.
SNA	7450	1	Skip on non-zero accumulator -- The content of each bit of the AC is sampled. If any bit contains a 1, the content of the PC is incremented by 1 to skip the next sequential instruction. If every bit contains a 0, the next instruction is executed.
SMA	7500	1	Skip on minus accumulator -- The content of AC0 is sampled. If AC0 contains a 1, indicating that the AC contains a negative 2's complement number, the content of the PC is incremented to skip the next sequential instruction. If AC0 contains a 0, the next instruction is executed.
SPA	7510	1	Skip on positive accumulator -- The content of AC0 is sampled. If AC0 contains a 0, indicating that the AC contains a positive 2's complement number (or 0), the content of the PC is incremented to skip the next sequential instruction. If AC0 contains a 1, the next instruction is executed.
CLA	7600	2	Clear accumulator -- Each bit of the AC is loaded with a binary 0.

Table 4-5 Microprogrammed Combinations of Group 2 Microinstructions

Mnemonic	Octal	Sequence	Description
SZA SNL	7460	1	Skip if AC=0 or L=1 or both.
SNA SZL	7470	1	Skip if AC≠0 and L=0.
SMA SNL	7520	1	Skip if AC<0 or L=1 or both.
SPA SZL	7530	1	Skip if AC=>0 and L=0.
SMA SZA	7540	1	Skip if AC=<0.
SPA SNA	7550	1	Skip if AC>0.
SMA SZA SNL	7560	1	Skip if AC=<0 or L=1 or both.
SPA SNA SZL	7570	1	Skip if AC>0 and L=0.
SZA CLA	7640	1,2	Skip if AC=0, then clear AC.
SNA CLA	7650	1,2	Skip if AC≠0, then clear AC.
SMA CLA	7700	1,2	Skip if AC<0, then clear AC.
SPA CLA	7710	1,2	Skip if AC>=0, then clear AC.

4.5.2.3 OSR Instruction -- The OSR instruction causes VT278 status bits to be ORed to AC<0:3> and AC<10:11>. The instruction format is shown in Figure 4-6.

If AC<0>=0, bits AC<1:3> define which foreign language keyboard is attached. The keyboard language is set at system assembly time by means of a four position DIP switch mounted on the VT278 processor module. The exact interpretation of AC<1:3> is a function of the keyboard translation ROM installed in the processor module. The three bits allow selection of one of seven foreign language keyboard formats. When AC<1:3>=000, standard US/UK is used but the display language can be changed to a foreign language by executing the select character set (SCS) ESCape sequence (Paragraph 5.2).

4.5.2.4 Group 3 Operate Instructions -- Group 3 operate instructions transfer data between the AC and multiplier quotient (MQ) registers. These microinstructions are intended primarily for use with external equipment, but they are also useful when the MQ is employed as a temporary storage register.

Figure 4-7 shows the format of a group 3 microinstruction. The operation code must be 7, to indicate an operate instruction, while bits 3 and 11 must both contain a 1, to indicate a group 3 microinstruction. Any one of bits 4, 5, or 7 may be set to

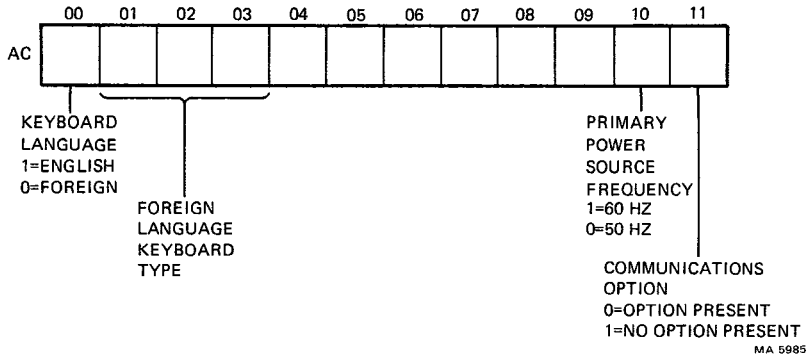


Figure 4-6 OSR Instruction Format

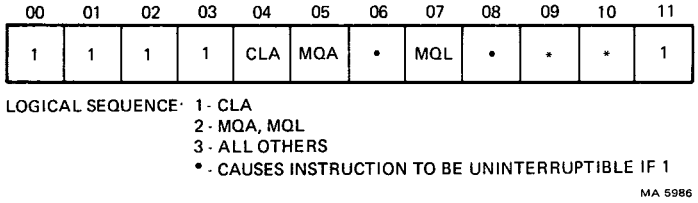


Figure 4-7 Group 3 Microinstruction Format

indicate a specific group 3 microinstruction. If more than one of these bits is set, the instruction is a microprogrammed combination of group 3 microinstructions. All group 3 microinstructions require 2.37 microseconds.

Table 4-6 lists the group 3 microinstructions, their assigned mnemonics, and the operations they perform.

4.5.3 Input/Output Operations

Input/output transfer (IOT) instructions have an opcode of 6. They are used to start the operation of peripheral devices and to transfer data between peripherals and the 6120. Three types of data transfer may be used to receive or transmit information between the 6120 and one or more peripheral I/O devices.

Programmed data transfer is a straightforward means of communicating with relatively slow I/O devices, such as the console terminal.

Program interrupt transfers use the interrupt system to service several peripheral devices simultaneously. On an intermittent basis, they permit computational operations to be performed concurrently with the data I/O operations. Both programmed data transfers and program interrupt transfers use the AC as a buffer, or storage area, for all data transfers. Since data may be transferred only between the AC and the peripheral, only one 12-bit word at a time may be transferred.

Table 4-6 Group 3 Operate Instructions

Mnemonic	Octal	Sequence	Description
NOP	74Ø1	3	No operation -- This instruction causes a delay in program execution, without affecting the state of the computer.
CLA	76Ø1	1	Clear accumulator -- Each bit of the AC is loaded with a binary Ø.
MQL	7421	2	Multiplier quotient load -- The content of the AC is loaded into the MQ. The AC is cleared, and the original content of the MQ is lost.
MQA	75Ø1	2	Multiplier quotient into accumulator -- The content of the MQ is combined with the content of the AC by a bitwise logical OR operation, and the result is loaded into the AC. The original content of the AC is lost, but the original content of the MQ is not affected. Note that this instruction provides the programmer with a direct inclusive OR operation.
SWP	7521	3	Swap accumulator and multiplier quotient -- The content of the AC and the content of the MQ are exchanged.
CAM	7621	3	Clear accumulator and multiplier quotient -- Each bit of both the AC and the MQ is loaded with a binary Ø.
ACL	77Ø1	3	Load AC with MQ -- The content of the MQ is load into the AC. The original content of the AC is lost. The original content of the MQ is not affected.
CLA SWP	7721	3	Clear accumulator, then swap -- The AC is cleared, then the MQ is loaded into the AC and ØØØØ is loaded into the MQ.

Direct memory access (DMA) transfers variable-size blocks of data between high-speed peripherals and memory with a minimum of program control by the 6120.

The IOT instruction format is shown in Figure 4-8. The first 3 bits, AC<0:2>, are always set to 6 to specify an IOT instruction. The next 6 bits, AC<3:8>, contain the device selection code that determines the specific I/O device for which the IOT instruction is intended. Device selection codes 00_8 and 20_8 through 27_8 specify internal operations, and may not be used by external devices. Internal operations are described in Paragraph 4.5.4. As many as 55 I/O devices can be specified. The last 3 bits, AC<9:11>, contain the code that determines the specific operation to be performed. The nature of this operation for any given IOT instruction depends entirely upon the circuitry designed into the I/O device interface.

4.5.3.1 Programmed Data Transfer -- Programmed data transfer is the easiest and most common means of performing data I/O. Every input/output transfer (IOT) instruction initiates one programmed transfer which may transmit data or status information either to or from a peripheral device. The amount of information that will be transferred by a single IOT instruction depends upon the particular operation that is coded into the instruction and the complexity of the I/O device interface. In general, programmed data transfers are limited to a maximum of 12 bits of data or 1 bit of status information per IOT instruction.

The computer operates much faster than most peripheral devices. For this reason, I/O routines must check the status of a device before executing an IOT instruction to be sure that the device is not still performing a previous operation. Device status is signalled through a system of 1-bit registers called flags. Every I/O device has a device flag which is set to 1 as soon as the device finishes a current operation and is ready to begin a new operation. If the flag is cleared (set to 0), the device is still performing the operation specified by the last IOT instruction received. Ready status is usually checked by means of a skip-on-flag IOT instruction, such that the computer does not skip out of a waiting loop until the I/O device is ready to begin a new operation.

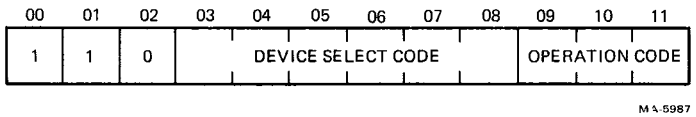


Figure 4-8 IOT Instruction Format

In general, for each I/O device there are at least three instructions:

1. An instruction to transfer information and/or operate the device.

2. An instruction to test the ready status of the device and skip on the ready (or not ready) status of the device.

3. An instruction to clear or set the device flag.

4.5.3.2 Program Interrupt Transfer -- The program interrupt system may be used to initiate programmed data transfers in such a way that the time spent waiting for device status is greatly reduced. It also provides a means of performing programmed data transfers between the 6120 and peripheral devices while executing another program. This is done by isolating the I/O handling routines from the mainline program and using the interrupt system to be sure that these routines are entered only when an I/O device flag is set. This indicates that the device is actually ready to perform the next data transfer.

The interrupt system allows external conditions to interrupt the computer program (which must be in main memory). If no internal higher priority requests are outstanding and the interrupt system is enabled, the 6120 grants the device interrupt at the end of the current instruction. After an interrupt has been granted, the interrupt enable flip-flop in the 6120 is reset so that no more interrupts are acknowledged until the interrupt system is reenabled under program control.

A program interrupt is simply a conventional JMS instruction to memory location 0000 in field 0 (octal code 4000) which is built and executed by circuits in the processor. Three conditions must be satisfied before a program interrupt may occur:

1. The interrupt system must be enabled,

2. The interrupt inhibit flip-flop must be cleared,

3. The interrupt request line must be asserted.

During the next machine cycle, the processor turns off the interrupt system so that no further interrupts may occur until the current interrupt has been serviced. It then executes the hardware-generated JMS to memory location 0000. This causes the address of the instruction that was due to be executed when the program interrupt occurred to be stored in location 0000 of memory field 0. The PC is then loaded with 0001, so that the instruction stored in location 0001 of field 0 is the next instruction to be executed.

This instruction is usually a jump to the starting address of an interrupt service routine, which generally performs all of the following operations:

1. Save the content of any registers (AC, MQ, link) that will be used by the interrupt service routine.
2. Determine which peripheral device caused the program interrupt.
3. Determine why the device caused an interrupt.
4. Correct the condition that resulted in the interrupt.
5. Restore the content of registers that were used by the interrupt service routine.
6. Turn the interrupt system on.
7. Resume mainline execution, usually by executing a JMP I Ø instruction (octal code 54ØØ).

A running program maintains control over the interrupt system by executing the processor IOT instructions described in Table 4-7.

Table 4-7 Main Memory Control Instructions

Mnemonic	Octal	Execution Time (us)	Description
SKON	6ØØØ	2.76	Skip if interrupt system on -- The state of the interrupt enable flip-flop is tested. If this flip-flop is set, indicating that the interrupt system is enabled, the PC is incremented to skip the next sequential instruction and the interrupt system is turned off in the same manner as by an IOF instruction.
ION	6ØØ1	2.37	Interrupt turn on -- The next program instruction is executed, then the interrupt system is enabled. Delaying the interrupt enable in this manner gives the interrupt service routine time to resume background program execution (by means of a JMP I Ø instruction) before another program interrupt occurs.

Table 4-7 Main Memory Control Instructions (Cont)

Mnemonic	Octal	Execution Time (us)	Description
IOF	6ØØ2	2.37	**Interrupt turn off** -- The interrupt system is disabled inhibiting further program interrupts, including any interrupt request that might have occurred during execution of the IOF instruction.
SRQ	6ØØ3	2.76	**Skip on interrupt request** -- The state of the interrupt request line is tested. If this line is asserted, indicating that one or more devices are requesting a program interrupt, the PC is incremented to skip the next sequential instruction.
GTF	6ØØ4	3.55	**Get flags** -- The following machine states are read into the indicated bits of the accumulator: ACØ Link AC1 GT flag AC2 Interrupt request line (1 if low; Ø if high) AC3 Power On flag AC4 1 AC5 Ø AC<6:8> ISF<Ø:2> AC<9:11> DSF<Ø:2>
RTF	6ØØ5	3.16	**Return flags** -- This instruction is the converse of GTF. Link is loaded from ACØ, GT flag is loaded from AC1, AC<6:8> are loaded into IB, AC<9:11> are loaded into DF, interrupt inhibit flip-flop is set, and bit 4 is loaded into the interrupt enable flip-flop. The AC is then cleared.
SGT	6ØØ6	2.76	**Skip if greater than** -- If the GT flag is set, the PC is incremented to skip the next sequential instruction.

Table 4-7 Main Memory Control Instructions (Cont)

Mnemonic	Octal	Execution Time (us)	Description
CAF	6ØØ7	2.76	**Clear all flags** -- Clear the AC, link, and GT flag. Clear the interrupt enable flip-flop and all peripheral flags.
PRØ	62Ø6	32Ø (8Ø-column) 324 (132-column)	**Panel request** -- Used to access locations in the display buffer memory. The location to be accessed is specified in row and column format in the two locations following the instruction itself: CALL, PRØ ROW /The physical /ROW number COL /The physical /COLUMN number Execution of the instruction causes immediate entry to panel memory where the row and column specified are used to access the display buffer memory. Return is made to CALL+3 with the contents of that memory location in the AC.
PR1	6216	Variable	**Panel request** -- Used to transfer blocks of data directly to the display buffer memory. Achieves a much faster screen update than normal terminal I/O instructions. This method should be used with caution. Like large updates the CPU remains in control panel memory for long periods of time. During this time, main memory interrupts are not serviced. Data to be put in the display buffer memory follows the instruction itself in the following format: UPDATE, PR1 ROW /This three word COL /group repeated DATA /for as many

Table 4-7 Main Memory Control Instructions (Cont)

Mnemonic	Octal	Execution Time (us)	Description
			.... /screen locations /as required. 7777 /Terminates /update /sequence. Executing this instruction causes immediate entry to panel memory where the locations in the display buffer memory specified by row, column are loaded with DATA. This continues until the 7777 terminator is reached when return is made to main memory to the location following the 7777. The screen may be filled by this method at speeds equivalent to a conventional terminal running at over 100K baud. The only limitation to the amount of data that may be passed at one time is that the instruction cannot cross field boundaries. Thus, the maximum number of screen locations that can be changed at one time is approximately 1300. An update of this length would take about 30 ms. As the display buffer memory is accessed directly by the PR1 instruction, certain sequences of normal terminal output executed just before the PR1 may not execute correctly. A PR1 should not be issued until at least 40 ms after any clear screen or line ESCape sequence has been transmitted by a normal TLS I/O sequence. To simplify executing such sequences the following method of finding out when the Clear function has completed may be used:

Table 4-7 Main Memory Control Instructions (Cont)

Mnemonic	Octal	Execution Time (us)	Description
			Send erase ESCape sequence. Send device status report (DSR) ESCape sequence and wait for cursor position report (CPR) ESCape sequence response. When CPR is received, the Erase function has been completed and it is then possible to execute the PR1 screen update.
PR2	6226	Indeterminate	**Panel request** -- Simulates entry into SET-UP mode and pressing one of keys Ø--9. Its use is: CALL, PR2 "N" /7 or 8 bit /representation /of Ø--9 The effect is as though SET-UP had been entered by pressing CTRL and SET-UP and then one of the keys Ø--9. As in SET-UP mode, only valid numbers have any effect. An invalid N causes a return to CALL+2.
PR3	6236	Not applicable	**Panel request** -- Used to access certain control panel memory routines from main memory. The effect of issuing this instruction is to transfer program execution immediately to that address in control panel memory which is defined in the memory location following the PR3 instruction:: PR3 nnnn /program exeuction /continues at address /nnnn in control panel /memory

Table 4-7 Main Memory Control Instructions (Cont)

Mnemonic	Octal	Execution Time (us)	Description
			This instruction should only be used with extreme caution. A jump to an arbitrary address in control panel memory could have unpredictable results. Certain routines in control panel memory are written as subroutines and may be accessed via the PR3 instruction. The routines are: Set TLS buffer size (Paragraph 4.5.4.3) and load/read keyboard LED pattern (Paragraph 4.5.4.4).
WSR	6246	2.76	I/O write -- Clears the AC.
GCF	6256	3.55	Get current fields -- The following bits are loaded into the AC: AC0 Link AC1 GT flag AC2 Interrupt request line (1 if low; 0 if high) AC3 Power on flag AC4 Interrupt enable flip-flop AC5 0 AC<6:8> Instruction field (IF<0:2>) AC<9:11> Data field (DF<0:2>)

4.5.3.3 Direct Memory Access Transfer -- Direct memory access, sometimes called data break, is the preferred form of data transfer to use with high-speed storage devices such as magnetic disk. The DMA mechanism transfers data directly between memory and peripheral devices. The 6120 is involved only in setting up the transfer; the transfers take place with no 6120 intervention on a "cycle stealing" basis. The DMA transfer rate is limited only by the bandwidth of the memory and the data transfer characteristics of the device.

The RL278 is the only DIGITAL device available for the VT278 that includes a DMA interface. The RL278 is supplied with software subroutines that perform all the necessary functions required for DMA operation.

4.5.4 Internal Operations

Internal operations are those IOT instructions which have a device selection code of 00_8 or 20_8 through 27_8. For the VT278, internal operations include:

Main memory control instructions,
Memory extension instructions, and
Stack operation instructions.

4.5.4.1 Main Memory Control Instructions -- Main memory control instructions are described in Table 4-7. They are primarily used to control the interrupt system and panel memory.

4.5.4.2 Memory Extension Control Instructions -- The internal memory extension control extends the addressing capabilities of the VT278 from 4096 words up to 32,768 words. It does so by appending three high-order bits to the memory address to define the memory field. The instructions described in Table 4-8 access extended memory fields by using the following registers to define the bits of addressing.

1. Instruction field (IF) - The IF is a 3-bit register that serves as an extension of the PC. The contents of the IF determine the field from which all instructions are taken and the field from which operands are taken in directly addressed AND, TAD, ISZ, or DCA instructions.

2. Data field (DF) - This 3-bit register determines the memory field from which operands are taken in indirectly addressed AND, TAD, ISZ, or DCA instructions and external IOTs, OSR, and WSR functions.

3. Instruction buffer (IB) - The IB serves as a 3-bit input buffer for the instruction field register. All field number transfers into the instruction field register are made through the IB.

4. Instruction save field (ISF) - When an interrupt occurs, the contents of the IF are loaded into the ISF. The IB can be restored by an RMF instruction before the exit from the interrupt subroutine.

5. Data save field (DSF) - When an interrupt occurs, the contents of the DF are loaded into the DSF. The DF can be restored by an RMF instruction before the exit from the interrupt subroutine.

Table 4-8 Memory Extension Control Instructions

Mnemonic	Octal	Execution Time (us)	Description
CDF	62N1	2.37	Change to data field N -- Loads the data field (DF) with the program-selected field number (N=Ø--7). All subsequent memory requests for operands of indirectly-addressed AND, TAD, ISZ, or DCA instructions and external IOTs, OSR, and WSR functions are automatically switched to that field.
CIF	62N2	2.37	Change to instruction field N -- Loads the instruction buffer (IB) with the program-selected field number (N=Ø--7) and inhibits processor interrupts until the next JMP or JMS instruction. At the end of either of these instructions, the contents of the instruction buffer register are transferred into the instruction field register (IF).
CDF CIF	62N3	2.37	Change to data field N and instruction field N -- Performs the combination of CDF and CIF operations.
RDF	6214	2.37	Read instruction field -- ORs the contents of the data field (DF) into AC<6:8>. All other bits of the AC are unaffected.
RIF	6224	2.37	Read instruction field -- ORs the contents of the instruction field (IF) into AC<6:8>. All other bits of the AC are unaffected.
RIB	6234	2.76	Read interrupt buffer -- ORs the contents of the instruction save field (ISF) and data save field (DSF) -- which are loaded from the IF and DF during a program interrupt -- into AC<6:8> and AC<9:11> respectively. Thus, AC<6:11> contains the IF and DF that were in use before the last program interrupt.

Table 4-8 Memory Extension Control Instructions (Cont)

Mnemonic	Octal	Execution Time (us)	Description
RMF	6244	2.37	Restore memory fields -- Restores the contents of the ISF and DSF (loaded from the IF and DF during a program interrupt) into the IB and DF, respectively. This command is used upon exit from the program interrupt subroutine in another field. The RMF instruction sets the inhibit interrupt flip-flop. At the time of the next JMP or JMS instruction, the IB is loaded into the IF and the interrupt inhibit flip-flop is cleared.

4.5.4.3 Set TLS Buffer Size (PR3 Instruction) -- At power-on, the size of the temporary buffer used to store characters transmitted from the user program that will be placed in the display buffer at the next End of Frame Panel Interrupt is set to 15 characters. The number may be changed to 29 characters, which allows faster screen updates.

```
PR3
5774    /set buffer size short (15)
....    /program execution continues here
        /after the buffer size is changed
        /by the CP memory subroutine.

PR3
5775    /set buffer size long (29)
....    /program execution continues here
        /after the buffer size is changed
        /by the CP memory subroutine.
```

4.5.4.4 Load/Read Keyboard LED Pattern (PR3 Instruction) -- In certain cases it is useful to be able to load (set) the pattern in the rightmost five keyboard LEDs directly.

```
PR3
2376    /set the LED pattern
....    /program execution continues here
        /after the LED pattern is changed.
```

The LED pattern must already be in AC<7:11> before executing the PR3 instruction. The AC is unchanged by this instruction. It is also possible to read the pattern in the rightmost five keyboard LEDs.

```
PR3
2374      /read the LED pattern
          /program execution continues here
          /after the LED pattern is read.
```

When program execution returns to main memory, the current LED pattern is contained in AC<7:11>.

4.5.4.5 Stack Instructions -- Certain IOT instructions are internally decoded to perform stack operations using internal stack pointers SP1 and SP2. The stacks are located in field Ø of main memory. Two separate stacks may be maintained -- one for the PC, the second for the AC. An increment of the stack pointer is defined as a pop off the stack. The stack operations defined are Push PC+1 (PPC1 or PPC2), Push AC (PAC1 or PAC2), Pop to PC (RTN1 or RTN2), Load AC from stack (POP1 or POP2) Read Stack Pointer (RSP1 or RSP2) and Load Stack Pointer (LSP1 or LSP2). These instructions are detailed in Table 4-9. (SP) is the contents of SP1 for PPC1, PAC1, RTN1, RSP1, or LSP1; otherwise, it is the contents of SP2.

The stack pointer moves down as items are added (pushed) to the stack and moves up as items are removed (popped). Therefore, it always points to the top of the stack.

Table 4-9 Stack Instructions

Mnemonic	Octal	Execution Time (us)	Description
PPC1	62Ø5	3.95	Push (PC+1) to stack 1
PPC2	6245	3.95	Push (PC+1) to stack 2
			The contents of the PC+1 are loaded into the address pointed to by the stack pointer. The content of the stack pointer is decremented to point to the new top of the stack.
			Operation: ((SP))<=(PC)+1 (SP)<=(SP)-1.
PAC1	6215	3.95	Push AC to stack 1
PAC2	6255	3.95	Push AC to stack 2
			The contents of the AC are loaded into the address pointed to by the stack pointer. The content of the stack pointer is decremented to point to the new top of the stack. The AC is unchanged by this instruction.

Table 4-9 Stack Instructions (Cont)

Mnemonic	Octal	Execution Time (us)	Description
			Operation: ((SP))<=(AC) (SP)<=(SP)-1.
RTN1	6225	3.55	Pop top of stack 1 to PC
RTN2	6265	3.55	Pop top of stack 2 to PC The stack pointer is incremented to point to the last address pushed. The content of this location is loaded into the PC. Operation: (SP)<=(SP)+1 (IF)<=(IB) (IIFF)<=Ø ; Interrupt inhibit ; flip-flop is ; cleared. (PC)<=((SP)).
POP1	6235	3.55	Pop top of stack 1 to AC
POP2	6275	3.55	Pop top of stack 2 to AC The stack pointer is incremented to point to the last pushed address. The content of this location is loaded into the AC. Operation: (SP)<=(SP)+1 (AC)<=((SP)).
RSP1	62Ø7	2.37	Read stack pointer 1 to AC
RSP2	6227	2.37	Read stack pointer 2 to AC The contents of the stack pointer is loaded into the AC. Operation: (AC)<=(SP).
LSP1	6217	2.37	Load stack pointer 1 from AC
LSP2	6237	2.37	Load stack pointer 2 from AC The contents of the AC is loaded into the stack pointer and the AC is cleared. Operation: (SP)<=(AC) (AC)<=ØØØØ.

4.6 PROGRAMMING CONSIDERATIONS

The following paragraphs describe the normal and vector modes for servicing interrupts.

4.6.1 Programming Modes

The VT278 I/O controller operates on device codes Ø3, Ø4, 32, 33, and 13, and is capable of operating in a type of vectored interrupt mode. Under certain circumstances it may be advantageous to modify existing programs to increase system performance. The interrupt system can, therefore, operate in either a normal mode or a vectored interrupt mode. The IOT instruction CMOD (613Ø) is used to change modes.

```
CMOD with ACØ=1 -- set interrupt mode normal
CMOD with ACØ=Ø -- set interrupt mode vectored
```

The interrupt system is in normal mode when power is turned on.

4.6.1.1 Normal Mode -- In this mode, interrupts force a subroutine call to location ØØØØ. Any skip instruction that tests a flag and finds it set clears the flag.

4.6.1.2 Vector Mode -- Vector mode can only be entered by issuing the IOT CMOD with ACØ=Ø. When operating with the interrupt system enabled, vector mode may allow faster interrupt identification. The normal interrupt skip chain is replaced by any sequence where the first IOT issued following the interrupt is of the form 6ABC (where C is anything other than 1 and AB is any unused device code). The result of issuing this IOT is that the I/O device causing the interrupt, or the highest priority device in the case of more than one device interrupting, returns its device code to AC<6:11> and the flag of that device is cleared. Thus the interrupt identification code could look something like:

```
IDENT,   6777          /Unused IOT to get vector
         TAD (PTR      /Start of jump table
         DCA TEM
         TAD I TEM     /Service routine start address
         PAC1          /Push AC to stack
         RTN1          /Go to service routine
```

4.6.1.3 Vector Mode Priority Sequence -- The interrupt priority is set during self-test and is:

Priority	I/O Device	Device Code
Highest	Comm. line #1 receive	3Ø
	Comm. line #2 receive	34
	Comm. line #1 send	31
	Comm. line #2 send	35
	Comm. option control/status	36
	Keyboard	Ø3
	Screen	Ø4
	Printer port input	32
Lowest	Printer port output	33

The priority setup code first determines whether the communications option is present or not. In the absence of the option, the keyboard (Ø3) has highest priority.

4.6.1.4 Vector Mode Caveats -- Since the interrupting device code is returned by executing a nonfunctional VT278 IOT, existing programs should be examined before using vector mode to be sure that no IOTs can be issued that will cause the I/O controller to respond when no response is expected. An example is an existing program that runs on a system with Power Fail; the Skip on Power Low IOT is 61Ø2. If this IOT were issued on a VT278 running in vector mode as the first IOT following an interrupt, the highest device would load its device code into the AC and its flag would be cleared.

CHAPTER 5
PROGRAMMING STANDARD I/O DEVICES

5.1 INTRODUCTION

This chapter contains programming information for the following standard devices integrated into the VT278.

Device	Device Code
Keyboard	Ø3
Display	Ø4
Printer port	32 and 33
Real-time clock	13
RX278 diskette subsystem interface	75

In addition to the information for programming these devices, Paragraph 5.6 contains a number of programming considerations regarding performance and interrupt servicing.

5.2 KEYBOARD AND DISPLAY INSTRUCTIONS

The keyboard and display on the VT278 can be controlled with conventional terminal I/O instructions (Paragraph 5.2.1), panel request (PR) instructions (Paragraph 4.5.4), and ESCape sequences (Paragraph 5.2.3). However, the keyboard/display is not a separate device as on most small computers. For the VT278, the keyboard/display device is emulated in control panel memory.

5.2.1 Terminal I/O Instructions

Tables 5-1 and 5-2 describe the terminal I/O instructions.

5.2.2 Terminal Emulation

Terminal emulation for the VT278 is accomplished by firmware in CP memory. The following paragraphs describe emulation for both display and keyboard operation.

5.2.2.1 Display Operation -- When a TLS instruction is executed in main memory, a CP interrupt is generated. Since the CP interrupt is serviced before the next main memory instruction can be executed, the character code present in the AC prior to the TLS instruction is still available in the AC and can be examined by a service routine in CP memory.

First, the character code is examined to see whether it is a normal character or if it has any special characteristics. If it is a normal, displayable character, it is stored in a temporary input buffer in a read/write portion of CP memory. Codes with

Table 5-1 Keyboard I/O Instructions

Mnemonic	Octal	Execution Time (us)	Description
KCF	6030	3.55	Sets keyboard flag (TTI).
KSF	6031	3.55	Skip on keyboard flag -- If keyboard flag was set since this IOT was last issued, the next instruction is skipped. Keyboard flag is set when a key is pressed. If skip occurs, keyboard flag is cleared.
KCC	6032	3.55	Clear AC.
KRS	6034	3.95	OR keyboard buffer to AC -- Contents of keyboard buffer are inclusively ORed with AC<5:11> and results are left in AC<5:11>. Contents of AC<0:4> are unchanged.
KIE	6035	3.55	Keyboard interrupt enable -- Enables/disables main memory interrupt by keyboard flag according to content of the AC11: AC11=0 -- clear interrupt enable; AC11=1 -- set interrupt enable.
KRB	6036	3.95	Load keyboard buffer to AC -- Performs combined operations of KCC and KRS. AC is cleared and contents of keyboard buffer are ORed into AC<5:11>.

Table 5-2 Display I/O Instructions

Mnemonic	Octal	Execution Time (us)	Description
SPF	6040	3.55	Sets display flag (TTO).
TSF	6041	3.55	Skip on display flag -- If display flag was set since this IOT was last issued, the next instruction is skipped. If skip occurs, display flag is cleared.
TCF	6042	3.55	Clear AC.
TIE	6045	3.55	Display interrupt enable -- Enables/disables main memory interrupts by display flag according to content of AC11: AC11=0 -- clear interrupt enable; AC11=1 -- set interrupt enable.
TLS	6046	3.55	Load AC<4:11> to screen -- Transfers character in AC to display memory buffer and sets display flag. AC is unchanged.

special characteristics (e.g., an ESCape character) may require that several codes following the initial character be examined to form a display function. These functions are also stored in the temporary buffer and use the additional 4 bits of the 12-bit word to mark them as functions and not as 8-bit ASCII characters.

After the incoming character is processed, an IOT is issued in CP memory that sets the TTO flag and operation is returned to the main memory program. When the main memory program executes a TSF instruction, it finds the TTO flag set, clears the flag, and skips the next instruction to allow the program to proceed. If the main memory program was running under interrupts, the return from CP memory to main memory causes a main memory interrupt, and the normal skip chain sequence finds the TTO flag set.

Characters and display functions stored in the CP temporary input buffer are periodically transferred into the display buffer memory and the information in the display buffer memory is used for updating the display. This procedure occurs 50 or 60 times a second for approximately 1 to 1.5 ms and is initiated by a CP interrupt which has a higher priority than main memory interrupts. The screen update is followed by a keyboard scan (Paragraph 5.2.2.2) and then control is returned to the main memory program.

5.2.2.2 Keyboard Operation -- The keyboard is scanned at the end of the display buffer update. If the scanner finds a key down, the scan causes an interrupt into control panel memory where, after decoding, valid characters are passed to the user program by loading them, one at a time, into the TTI buffer. At the same time the TTI flag, which can be tested by the KSF instruction, is set. Control then returns to the main memory program.

When the main memory program executes a KSF instruction, it finds the TTI flag set, clears the flag, and skips the next instruction to allow the program to proceed. If the main memory program was running under interrupts, the return from CP memory to main memory causes a main memory interrupt and the normal skip chain sequence finds the TTI flag set. In either case, a KRB or KRS instruction reads the character from the TTI buffer to the AC. A KCC or KRS instruction causes another CP interrupt to indicate to the keyboard handling routine in CP memory that the TTI buffer is now free to accept another character when or if one becomes available.

5.2.3 ESCape Sequences For Keyboard/Display

The information in this paragraph describes the ESCape sequences for a VT278 equipped with a US/UK keyboard that does not contain foreign language support. Additional information for foreign language keyboards is included in Appendices A and B.

The VT278 has many control commands (ESCape sequences) that cause the keyboard/display to take action other than to display a character on the screen. In this way, a user program can transmit ESCape sequences that can move the cursor, change modes, etc. The ASCII codes for characters in all languages are shown in Appendix A.

The following example shows the ASCII octal representation of characters in an ESCape sequence that turns off all character attributes, and then turns on underscore and blink attributes. This sequence is called select graphic rendition (SGR).

Example

ESC	[	Ø	;	4	;	5	m	(SGR sequence)
Ø33	133	Ø6Ø	Ø73	Ø64	Ø73	Ø65	155	(ASCII octal equivalent)

In the following ESCape sequence descriptions, the symbols P_v, P_h, P_s, P_1, P_n, and F_x are parameters that represent numeric or alphabetic values. In addition, in those sequences that permit a string of parameters, a semicolon (;) is required to separate the parameters.

5.2.3.1 Cursor Position Report (CPR)

ESC [P_v ; P_h R

P_v = cursor vertical position (row)
P_h = cursor horizontal position (column)

CPR is transmitted to the user program in response to a DSR (device status report) sequence. CPR reports the active position of the cursor with values for parameters P_v and P_h. A response with no parameters or parameters of Ø;Ø is equivalent to the cursor at home position.

5.2.3.2 Cursor Backward (CUB)

ESC [P_n D

CUB moves the cursor to the left. The distance moved is determined by P_n. If P_n is Ø or 1, the cursor moves one position to the left. If P_n is n, the cursor moves n positions to the left. An attempt to move the cursor to the left of the left margin leaves the cursor at the left margin.

5.2.3.3 Cursor Down (CUD)

ESC [P_n B

CUD moves the cursor down without altering column position. The number of rows moved is determined by P_n. If P_n is Ø or 1, the cursor moves down one line. If P_n is n, the cursor moves n rows down. An attempt to move the cursor below the bottom margin stops the cursor at the bottom margin.

5.2.3.4 Cursor Forward (CUF)

ESC [P_n C

CUF moves the cursor to the right. The number of columns moved is determined by P_n. If P_n is Ø or 1, the cursor moves one position to the right. If P_n is n, the cursor moves n positions to the right. An attempt to move the cursor to the right of the right margin stops the cursor at the right margin.

5.2.3.5 Cursor Position (CUP)

ESC [P_v ; P_h H

P_v = cursor vertical position (row)
P_h = cursor horizontal position (column)

CUP moves the cursor to the position specified by P_v and P_h. A value of Ø or 1, or no value for P_v or P_h moves the cursor to the first row or column, respectively.

5.2.3.6 Cursor Up (CUU)

ESC [P_n A

CUU moves the cursor up without changing column position. The number of rows moved is determined by P_n. If P_n is Ø or 1, the cursor moves up one row. If P_n is n, the cursor moves up n rows. An attempt to move the cursor above the top margin stops the cursor at the top margin.

5.2.3.7 Device Attributes (DA)

ESC [P_n c

The user program sends DA with P_n equal to Ø or not present. DA is a request for the control panel memory emulation code to transmit the VT278 characteristics back to the user program in the following ESCape sequence.

ESC [? 8 ; P_s c (P_s = configuration-dependent parameter)

P_s	Configuration
Ø	Basic VT278
1	32K memory
2	Basic VT278 plus communication option
3	32K memory plus communication option

If the VT278 contains the foreign language support option, the ESCape sequence that responds to the DA request is extended as follows.

ESC [? 8 ; P_s ; P_l c

P_s is already defined in this paragraph and P_l indicates the keyboard language.

P_l	Keyboard Language
1	German
2	Dutch
3	French
4	French-Canadian

5.2.3.8 Erase in Display (ED)

ESC [P_n J

ED erases some or all characters on the screen according to P_n.

P_n	Function
Ø	Erase from current cursor position to end of screen, inclusive.
1	Erase from start of screen to current position, inclusive.
2	Erase entire screen.

In all cases, the cursor position does not change.

5.2.3.9 Erase in Line (EL)

ESC [P_n K

EL erases some or all characters in the current line according to P_n.

P_n	Function
Ø	Erase from current cursor position to end of line, inclusive.
1	Erase from start of line to current cursor position, inclusive.
2	Erase entire line.

In all cases the cursor position does not change.

5.2.3.1Ø Device Status Report (DSR)

ESC [6 r

DSR is a request from the user program for the emulation firmware to report the current cursor position via a CPR sequence (Paragraph 5.2.3.1).

5.2.3.11 Horizontal and Vertical Position (HVP)

ESC [P_v ; P_h f

P_v = cursor vertical position (row)
P_h = cursor horizontal position (column)

HVP is identical to CUP. It moves the cursor to the position specified by P_v and P_h. A value of Ø or 1, or no value for P_v or P_h moves the cursor to the first row or column, respectively.

5.2.3.12 Index (IND)

ESC D

IND moves the cursor down one row without changing column position. If the cursor is at the bottom margin, a scroll up is performed.

5.2.3.13 Next Line (NEL)

ESC E

NEL moves the cursor to the first position on the next row. If the cursor is at the bottom margin, a scroll up is performed.

5.2.3.14 Reverse Index (RI)

ESC M

RI moves the cursor up to the preceding row without changing horizontal position. If the cursor is at the top margin, a scroll down is performed.

5.2.3.15 Select Character Set (SCS)

ESC (F_x Define GØ set
ESC) F_x Define G1 set

These sequences allow the user program to predefine two alternate character sets as GØ and G1. Once defined, the character sets can be selected by transmitting SI ($Ø17_8$) for GØ or SO ($Ø16_8$) for G1.

F_x	Representation
A	United Kingdom Set
B	USASCII Set
Ø	Special Graphics
1	Alternate Character ROM
2	Alternate Character ROM, Special Graphics
3*	French-Canadian
4*	Dutch
K*	German
R*	French

* Only available on VT278 systems equipped with extended language option.

5.2.3.16 Select Graphic Rendition (SGR)

ESC [P_n ;....; P_n m

SGR establishes the attributes to be appended to subsequent characters sent to the screen according to the following values for P_n.

P_n	Action
Ø	All attributes off
1	Display at increased intensity
4	Display underscored
5	Display blinking
7	Display negative (reverse) image

There are four display intensity levels at a given brightness setting: Ø, 1, 2, and 3. (Where Ø is no display and 1, 2, and 3 are sequentially increasing levels of intensity.) Certain combinations of attributes result in displays as follows.

P_n	Character/Underscore Level	Background Level
Ø	2	Ø
1	3	Ø
4	2	Ø
5	2/1*	Ø
1;4	3	Ø
1;5	3/2*	Ø
4;5	2/1*	Ø
1;4;5	3/2*	Ø
7	Ø	2
1;7	Ø	3
4;7	Ø	2
5;7	Ø	2/1*
1;4;7	Ø	3
1;5;7	Ø	3/2*
4;5;7	Ø	2/1*
1;4;5;7	Ø	3/2*

* The display levels indicated alternate at the BLINK rate.

5.2.3.17 Keypad Application Mode (DECKPAM)

ESC =

DECKPAM allows the keys on the auxiliary keypad to transmit the following ESCape sequences:

Key	ESCape Sequence	Key	ESCape Sequence
Ø	ESC O p	-	ESC O m
1	ESC O q	,	ESC O l
2	ESC O r	.	ESC O n
3	ESC O s	ENTER	ESC O m
4	ESC O t	PF1	ESC O P
5	ESC O u	PF2	ESC O Q
6	ESC O v	PF3	ESC O R
7	ESC O w	PF4	ESC O S
8	ESC O x		
9	ESC O y		

In addition, cursor control keys transmit the following ESCape sequences:

Cursor Control Key		ESCape Sequence (Set Mode)*	ESCape Sequence (Reset Mode)*
Up	↑	ESC O A	ESC [A
Down	↓	ESC O B	ESC [B
Right	-->	ESC O C	ESC [C
Left	<--	ESC O D	ESC [D

* Paragraphs 5.2.3.19 and 5.2.3.20 describe set mode (SM) and reset mode (RM).

5.2.3.18 Keypad Numeric Mode (DECKPNM)

ESC >

In this mode, auxiliary keypad keys transmit codes that match the characters on the keys.

Key	Character	Key	Character
Ø	Ø	-	-
1	1	,	,
2	2	.	.
3	3	ENTER	Return (Ø15)
4	4	PF1	ESC O P
5	5	PF2	ESC O Q
6	6	PF3	ESC O R
7	7	PF4	ESC O S
8	8		
9	9		

Cursor control keys transmit cursor control codes.

Key		Sequence
Up	↑	ESC [1 A
Down	↓	ESC [1 B
Right	-->	ESC [1 C
Left	<--	ESC [1 D

5.2.3.19 Set Mode (SM)

ESC [? P_s ;....; P_s h

SM sets one or more modes as specified by each parameter in the parameter string. A mode is considered set until it is reset by a reset mode (RM) sequence.

P_s	Function	Mnemonic
1	Cursor key*	DECCKM
3	Set 132-column mode	DECCOLM
8	Set Auto-repeat	DECARM

* See keypad application mode, DECKPAM, Paragraph 5.2.3.1

5.2.3.20 Reset Mode (RM)

ESC [? P_s ;....; P_s l

RM resets one or more modes as specified by each parameter in the parameter string.

P_s	Function	Mnemonic
1	Cursor key*	DECCKM
3	Reset to 8Ø-column mode	DECCOLM
8	Reset to nonrepeating keys	DECARM

* See keypad application mode, DECKPAM, Paragraph 5.2.3.17

5.2.3.21 Save Cursor (DECSC)

ESC 7

DECSC saves the current cursor position, graphic rendition, and character set.

5.2.3.22 Restore Cursor (DECRC)

ESC 8

DECRC restores the previously saved cursor position, graphic rendition, and character set.

5.2.3.23 Print Screen (DECPRS)

ESC [i

DECPRS transfers the screen contents to the serial printer port. The VT278 remains in panel memory until this function is completed.

5.2.3.24 Load LEDs (DECLL)

ESC [P_s q

DECLL allows the user program to control the programmable LEDs on the keyboard.

P_s	Action
Ø	Turn OFF all five LEDs
1	Turn ON LED 1
2	Turn ON LED 2
3	Turn ON LED 3
4	Turn ON LED 4
5	Turn ON LED DSR

5.2.4 Keyboard/Display Programming Examples

The program in example 1 continues to accept characters until a dollar sign ($) is typed at the keyboard. It then stores ØØØØ in the next memory location and halts.

The program in example 2 types the characters whose ASCII codes were stored by the first program, and halts when a location with contents equal to zero is reached. Both programs use locations beginning at 2ØØØ as a storage buffer for the ASCII characters.

Example 1 Program to Accept ASCII Characters

```
*200
START,      CLA CLL          /CLEAR ACCUMULATOR AND LINK
            TAD BUFF         /SET UP BUFFER SPACE
            DCA BUFFPT       /AND INITIALIZE POINTER
LISN,       KSF              /KEYBOARD STRUCK YET?
            JMP .-1          /NO: CHECK AGAIN
            KRB              /YES: GET THE CHARACTER
            TLS              /ACKNOWLEDGE IT ON PRINTER
            DCA I BUFFPT     /STORE THE CHARACTER
            TAD I BUFFPT     /CHECK FOR TERMINAL
            TAD MDOLAR       /DOLLAR SIGN ($)
            SNA              /IS CHARACTER A "$"?
            JMP DONE         /YES: STORE ZERO OVER $
            ISZ BUFFPT       /NO: INCREMENT POINTER
            JMP LISN         /GET ANOTHER CHARACTER
DONE,       CLA CLL          /CLEAR ACCUMULATOR
            DCA I BUFFPT     /STORE ZERO IN LAST
            HLT              /BUFFER LOCATION AND HALT
BUFF        2000             /BUFFER BEGINS AT
BUFFPT,     0                /CORE LOCATION 2000
MDOLAR,     7534             /TWO'S COMPLEMENT OF 0244
$
```

Example 2 Program to Print Out ASCII Characters

```
*300
START,      CLA CLL          /CLEAR ACCUMULATOR AND LINK
            TLS              /RAISE PRINTER FLAG
            TAD BUFF         /SET UP BUFFER SPACE AND
            DCA BUFFPT       /INITIALIZE BUFFER POINTER
            JMS CRLF         /RETURN CARRIAGE
CHRTYP,     TAD I BUFFPT     /GET A CHARACTER
            SNA              /IS IT ALL ZEROS?
            HLT              /YES: HALT
            JMS TYPE         /NO: TYPE THE CHARACTER
            ISZ BUFFPT       /INCREMENT BUFFER POINTER
            JMP CHRTYP       /GET ANOTHER CHARACTER
CRLF,       0                /CRLF SUBROUTINE
            TAD K215         /GET ASCII CARRIAGE RETURN
            JMS TYPE         /PRINT IT
            TAD K212         /GET ASCII LINE FEED
            JMS TYPE         /PRINT IT
            JMP I CRLF       /AND RETURN
TYPE,       0                /TYPE SUBROUTINE
            TSF              /PRINTER READY YET?
            JMP .-1          /NO: CHECK AGAIN
            TLS              /YES: TYPE CHARACTER
            CLA CLL          /CLEAR ACCUMULATOR
            JMP I TYPE       /AND RETURN
BUFF,       2000             /BUFFER BEGINS AT
BUFFPT,     0                /CORE LOCATION 2000
K215,       215              /ASCII CARRIAGE RETURN
K212,       212              /ASCII LINE FEED
$
```

5.3 PRINTER INSTRUCTIONS

A 25-pin male connector at the rear of the VT278 allows you to use any one of a variety of serial printers (LQPØ2, LA34, etc.) that are RS232-compatible. When power is turned on to the VT278, the baud rate for the printer is initially set at 96ØØ baud. You can change the baud rate by using the TSB2 (IOT 6333) instruction. This instruction is described in Table 5-3 with the remaining printer instructions. Instructions with device code 33 are for output of data from the AC to the printer. Instructions with device code 32 are for inputting data from the printer to the AC.

Table 5-3 Printer Instructions

Mnemonic	Octal	Description
KCF2	632Ø	Set printer input flag.
KSF2	6321	Skip on printer input flag -- If printer flag changed from false to true since this IOT was last issued, the next instruction is skipped. If skip occurs, printer input flag is cleared.
KCC2	6322	NOP
KIE2	6325	Printer input interrupt enable -- Enables/disables interrupts by printer input flag according to content of AC11: AC11=Ø -- clear interrupt enable; AC11=1 -- set interrupt enable.
KRB2	6326	Transfer printer input data to AC<4:11> -- Contents of input buffer are transferred to AC<4:11> and printer input flag is set.
SPF2	633Ø	Set printer input flag (TTI2).
TSF2	6331	Skip on printer output flag -- If printer interface flag changed from false to true since the IOT was last issued, the next instruction is skipped. If skip occurs, printer output flag is cleared. This flag changes from false to true after printer interface is read by a 6336 instruction and interrupts if printer output interrupts are enabled.
TCF2	6332	Clear AC.
TSB2	6333	Set baud rate -- Printer port baud rate is set according to contents of AC<8:11>:

Table 5-3 Printer Instructions (Cont)

Mnemonic	Octal	Description			
		Conts	Rate	Conts	Rate
		Ø	5Ø	1Ø	18ØØ
		1	75	11	2ØØØ
		2	11Ø	12	24ØØ
		3	134.5	13	36ØØ
		4	15Ø	14	48ØØ
		5	3ØØ	15	72ØØ
		6	6ØØ	16	96ØØ
		7	12ØØ	17	192ØØ
TIE2	6335	Printer output interrupt enable -- Enables/disables interrupts by printer input flag according to content of AC11: AC11=Ø -- clear interrupt enable; AC11=1 -- set interrupt enable.			
TLS2	6336	Transfer AC<4:11> to printer -- Contents of AC<4:11> are transferred to printer buffer. Printer output flag is set when print operation is completed.			

5.4 REAL-TIME CLOCK INSTRUCTIONS

The real-time clock is crystal controlled and operates at 1ØØ Hz. It has a basic accuracy of ±Ø.Ø1 percent. Since the clock is crystal controlled, cumulative errors can be expected, but they will not exceed ±3 seconds in an 8 hour period.

The instructions described in Table 5-4 control the real-time clock.

5.5 RX278 DISKETTE SUBSYSTEM

The RX278 controller is an integral part of the VT278. Either one or two pairs of floppy disk drives can be connected to the controller. Drives can be RXØ1 (single density) or RXØ2 (double density) and any combination of RXØ1 and RXØ2 drives can be used. However, some DEC standard software may require that the drives be of the same type. RXØ2 drives can be programmed for either single or double density recording. RXØ1 drives will only operate in single density mode.

5.5.1 RX278 Diskette Subsystem Instruction Set

The RX278 controller allows two modes of data transfer between the VT278 and the floppy disk drives: 8-bit and 12-bit word lengths. For the RXØ1, in 12-bit mode, 64 words are written in a diskette sector, requiring two sectors to store one page of information. The diskette capacity in this mode is 128,128 12-bit words (1ØØ1 pages). In 8-bit transfer mode, 128 8-bit words are written in each sector. Diskette capacity is 256,256 8-bit words, which is a

Table 5-4 Real-Time Clock Instructions

Mnemonic	Octal	Description
CLLE	6135	Set/clear clock interrupt -- Enables/disables interrupts by clock: AC11=Ø -- clear interrupt enable; AC11=1 -- set interrupt enable.
CLCL	6136	Clear clock flag.
CLSK	6137	Skip on clock flag -- Senses clock flag and increments PC if set, thereby skipping next sequential instruction.

33 percent increase in diskette capacity over 12-bit mode. The RXØ2 has double size sectors with a capacity of 256,256 12-bit words, or 512,512 8-bit words. Since 12-bit mode does not fully pack the sectors with data, the hardware puts in extra Øs. Data transfer requests occur 23 us after the previous request was serviced for 12-bit mode (918 us for 8-bit mode). There is no maximum time between the transfer request from the floppy disk drive system and servicing that request by the processor. This allows the data transfer to and from the floppy disk drive system to be interrupted without loss of data.

The format for RX278 instructions is shown in Figure 5-1 and the error register, RX2ES, is shown in Figure 5-2. The instruction set is described in Table 5-5.

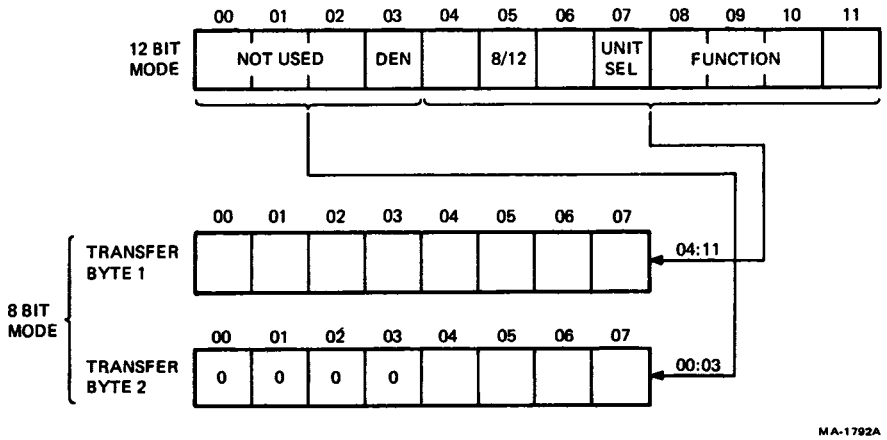


Figure 5-1 RX278 Diskette Subsystem Instruction Format

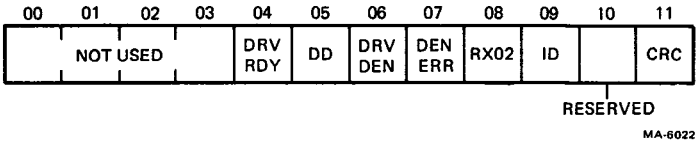


Figure 5-2 Error Status Register (RX2ES)

Table 5-5 RX278 Diskette Subsystem Instruction Set

Mnemonic	Octal	Description
SEL	6750	**Select RX01/RX02 drive pair** -- If AC<11>=1, select drive pair B. If AC<11>=0, select drive pair A. Drive pair A is selected by a CAF instruction or at power-on time.
LCD	6751	**Load command register** -- Must only be issued when DONE is true. The contents of AC are loaded into the interface register, and under timing control from RX01/RX02, sent serially to RX01/RX02. The AC is cleared following this IOT. AC bit 5 selects 8- or 12-bit mode of data transfer. 12-bit mode is set following CAF or power-on. AC bit 7 selects right or left drive of selected drive pair (0=left, 1=right). This IOT clears the DONE flag (DONE=false) and executes the function specified in AC bits 8, 9, and 10. The eight functions that can be performed are listed below, and explained in Paragraph 5.5.2. **AC<8:10> Function** 000 Fill buffer 001 Empty buffer 010 Write sector 011 Read sector 100 Change density 101 Read status 110 Write sector (deleted data) 111 Read error register

Table 5-5 RX278 Diskette Subsystem Instruction Set (Cont)

Mnemonic	Octal	Description
		For an RXØ2 in 8-bit mode, the Load command is stored in two 8-bit transfers. The first 8 bits of the command word (bits 4--11 in Figure 5-1) are stored. Then transfer request (TR) is asserted and an XDR is performed to transfer the remaining bits of data (bit 3, DEN, and bit 2, Figure 5-1) right-justified. Extra bits in the second 8-bit transfer are filled with zeros. Upon completing the transfer of the second 8-bit byte, DONE is asserted to end the function.
XDR	6752	Transfer data -- Data is transferred between the AC and the interface register. When the DONE flag is cleared (false) the direction of transfer depends on the current function. When the DONE flag is set (true) transfer is always to the AC. DONE=false; transfer to AC. 12-bit mode: the AC is cleared, then the contents of the interface register are transferred to the AC<Ø:11>. 8-bit mode: the contents of the interface register are inclusive ORed with the contents of the AC<4:11>. The result is placed in AC<4:11>. Bits AC<Ø:3> are unchanged. DONE=false; transfer to interface register. 12-bit mode: the contents of the AC<Ø:11> are transferred to the interface register. The AC is unchanged. 8-bit mode: the contents of AC<4:11> are transferred to the interface register. The AC is unchanged. DONE=true; transfer to AC. (Same as DONE=false; transfer to AC).

Table 5-5 RX278 Diskette Subsystem Instruction Set (Cont)

Mnemonic	Octal	Description
STR	6753	Skip on transfer request -- If the transfer request (TR) flag from RXØ1/RXØ2 changed from false to true since this IOT was last issued, the next instruction is skipped. If skip occurs, the flag is cleared. This IOT does not change AC. The TR flag should be tested prior to transferring data or address words with the XDR instruction to be sure data or address has been received or transferred, or after an LCD instruction to be sure the command is in the interface register. When an XDR follows an LCD, the flag only has to be tested once between the two instructions.
SER	6754	Skip on error -- If the ERROR flag from the RXØ1/RXØ2 changed from false to true since this IOT was last issued, the next instruction is skipped. If skip occurs, the ERROR flag is cleared. This IOT does not change the AC. Floppy disk drive system error status register (RX2ES) (Figure 5-2) contains the current error and status conditions of the selected drive. This read-only register can be accessed by the read status function (1Ø1). RX2ES is also available in the interface register upon completion of any function. RX2ES is accessed by the XDR instruction. Error bit meanings are listed in Table 5-6.
SDN	6755	Skip on done -- If the DONE flag from the RXØ1/RXØ2 changed from false to true since this IOT was last issued, the next instruction is skipped. If skip occurs, the DONE flag is cleared. This flag will interrupt if interrupts are enabled.

Table 5-5 RX278 Diskette Subsystem Instruction Set (Cont)

Mnemonic	Octal	Description
INTR	6756	Interrupt enable -- Enables/disables interrupts by DONE flag according to the contents of AC11: AC11=Ø -- clear interrupt enable; AC11=1 -- set interrupt enable. DONE flag interrupts are disabled by CAF instruction and following power-on.
RXINIT	6757	Initialize RXØ1/RXØ2 -- The following are cleared: DONE flag Transfer request flag ERROR flag Interrupt enable Unit select set to A. All drives connected are caused to initialize. At the end of the operation the DONE flag is set. The AC is not changed by this IOT.

Table 5-6 Error Status Register (RX2ES) Bit Functions

Bit	Description
11	CRC error -- Cyclic redundancy check at end of data field has indicated an error. Data must be considered invalid. It is suggested that data transfer be retried up to 10 times, as most data errors are recoverable (soft).
10	Reserved.
9	Initialize done -- Indicates completion of initialize routine. Can be asserted due to RX02 power failure, system power failure, or programmable or bus initialize. Bit is not available within RX2ES from a read status function.
8	RX02 -- Asserted if RX02 system being used.
7	DEN ERR -- Indicates that density of function does not agree with drive density. Upon detection of this error control terminates operation and asserts ERROR and DONE.
6	DRV DEN -- Indicates density of diskette in drive selected (0=single, 1=double).
5	Deleted data (DD) -- While reading data, a deleted data mark was detected in the identification field. Data following will be collected and transferred normally, since deleted data mark has no further significance within RX02. Any alteration of files or actual deletion of data due to this mark must be done by user software. This bit will be set if a successful or unsuccessful write deleted data function is performed.
4	Drive ready -- Asserted if unit currently selected exists, is properly supplied with power, has a diskette installed properly, has its door closed, and has a diskette up to speed.

5.5.2 Load Command Function Codes

RX01/RX02 functions are initiated by means of the Load command described in Table 5-5. The done flag should be tested and cleared with the SDN instruction in order to verify that the RX01/RX02 is in the Done state prior to issuing the command instruction. Upon receiving a command instruction while in the Done state, the RX01/RX02 enters the Not Done state while the command is decoded. Each of the eight functions summarized below requires that a strict protocol be followed for the successful transfer of data, status, and address information. The protocol for each function is described in the following paragraphs.

5.5.2.1 **Fill Buffer (ØØØ)** -- Fill buffer loads the floppy disk drive system sector buffer from the processor with 64 words in 12-bit mode or 128 words in 8-bit mode for the RXØ1 and 128 words in 12-bit mode or 256 words in 8-bit mode for the RXØ2. This instruction only loads the sector buffer. To complete the transfer to the diskette, the write sector function must then be performed. The buffer may also be read back by means of the empty buffer function to verify the data.

Upon decoding the fill buffer function, the floppy disk drive system sets the transfer request (TR) flag signaling a request for the first data word. The TR flag must be tested and cleared by the processor with the skip on transfer request (STR) instruction prior to each successive XDR IOT. The data word can then be transferred to the interface register by means of the XDR IOT. The floppy disk drive system then moves the data word from the interface register to the sector buffer and sets the TR flag as a request for the next data word. This sequence is repeated until the sector buffer has been loaded. After the last word has been loaded into the sector buffer, the content of the error status register (RX2ES) moves to the interface register and the floppy disk drive system sets the done flag to indicate the completion of the function. The processor does not have to count the data transfers. Any XDR command, after Done is set, loads the RX2ES status word into the AC. The sector buffer must be completely loaded before the interface sets the done flip-flop and recognizes a new command. An interrupt would now occur if the interrupt enable flip-flop were set.

5.5.2.2 **Empty Buffer (ØØ1)** -- This function moves the contents of the sector buffer to the processor. When the empty buffer function is decoded, RX2ES bit 11 is cleared, and the TR flag is set with the first data word in the interface register. This TR flag requests a data transfer from the buffer to the processor. The flag must be tested and cleared, then the word can be moved to the AC by an XDR command. The direction of transfer for an XDR command is controlled by the floppy disk drive system. The TR flag is set again with the next word in the interface register. This sequence is repeated until all words have been transferred, thus emptying the sector buffer. The done flag is set after the RX2ES moves into the interface register indicating the end of the function. An interrupt would now occur if the interrupt enable flip-flop were set.

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

5.5.2.3 **Write Sector (Ø1Ø)** -- This function transfers the contents of the sector buffer to a specific track and sector on the diskette. Upon decoding this function, the interface clears bit 11 of the RX2ES and sets the TR flag, signifying a request for the sector address. The TR flag must be tested and cleared before the sector address can be loaded into the interface register by the XDR command.

The TR flag is set again, signifying a request for the track address. The TR flag must be tested and cleared; then the track address may be loaded into the interface register by the XDR command.

The floppy disk drive system tests the supplied track address to determine if it is within allowable limits. If not, the RX2ES moves to the interface register, the error and done flags are set, and the function is terminated.

If the track address is legal, the floppy disk drive moves the head of the selected drive to the selected track, locates the requested sector, transfers the contents of the sector buffer and a CRC character to that sector, and sets the done flip-flop. Any errors encountered in the seek operation cause the function to cease, the RX2ES to be loaded into the interface register, and the error and done flags to be set. If no errors are encountered, the RX2ES is loaded into the interface register and only the done flag is set.

NOTE: The write sector function does not destroy the contents of the sector buffer.

5.5.2.4 Read Sector (Ø11) -- This function moves a sector of data from a specified track and sector on the floppy disk to the sector buffer. Upon decoding this function, the RXØ1/RXØ2 clears RX2ES bits 5 and 11 (deleted data and CRC error, respectively) and sets the TR flag signifying the request for the sector address. The flag must be tested and cleared. The sector address is then loaded into the interface register by the XDR command. The TR flag is set again signifying a request for the track address. The flag is tested and cleared by the processor, and the track address is then loaded into the interface register by an XDR command. The legality of the track address is checked by the floppy disk drive system. If illegal, the error and done flags are set when RX2ES moves to the interface register, and the function is terminated. Otherwise, the floppy disk drive system moves the head to the specified track, locates the specified sector, transfers the data to the sector buffer, and computes and checks CRC for data. If no errors occur, the done flag is set with the RX2ES in the interface register. If an error occurs anytime during execution of the function, the function is terminated by setting the error and done flags with RX2ES in the interface register. Detection of a CRC error results in RX2ES bit 11 being set. If a deleted data mark was encountered at the beginning of the desired data field, RX2ES bit 5 is set.

5.5.2.5 Change Density (1ØØ) -- This function operates as a NOP for RXØ1 drives. For RXØ2 drives, the new density is selected using AC bit 3: Ø = single density; 1 = double density.

5.5.2.6 Read Status (101) -- Upon decoding this function, the floppy disk drive system moves the RX2ES to the interface register and sets the done flag. The RX2ES can then be read by the XDR.

NOTE: The average time for this function is 250 ms. Excessive use of this function results in substantially reduced throughput.

5.5.2.7 Write Deleted Data Sector (110) -- This function is identical to the write data function except that a deleted data mark is written prior to the data field rather than the normal data mark. RX2ES bit 5 (deleted data) will be set in the interface register upon completion of the function.

5.5.2.8 Read Error Register Function (111) -- The read error register function can be used to retrieve explicit error information upon detection of the error flag. Upon receiving this function, the floppy disk drive system moves an error code to the interface register and sets the done flip-flop. The interface register can then be read via an XDR command and the code interrogated to determine which type of failure occurred.

Table 5-7 lists the read error register functions.

Table 5-7 Read Error Register Codes

Octal Code	Error Code Meaning
0010	Drive 0 failed to see home on initialize.
0020	Drive 1 failed to see home on initialize.
0040	Tried to access track greater than 76.
0050	Home found before desired track reached.
0070	Desired sector not found after looking at 52 headers (2 revolutions).
0110	More than 40 us and no SEP clock seen.
0120	Preamble could not be found.
0130	Preamble found but no ID mark found within allowable time span.
0150	Header track address of good header does not compare with desired track.
0160	Too many tries for an IDAM (identifies header).
0170	Data AM not found in allotted time.
0200	CRC error on reading sector from disk.
0220	R/W electronics failed maintenance mode test.
0240	Density error.
0250	Wrong key word for set media density command.

5.5.3 Programming Examples (RX01/RX02)

The following programming examples demonstrate a technique for reading data from a sector into the floppy disk interface buffer (example 1), and then transferring the data into main memory (example 2).

Floppy Disk Programming Example 1

```
/READ SECTOR EXAMPLE

            TAD (FUNCT      /FUNCTION = 000 000 000 110
                            /(bits <8:10> = 011)
            LCD             /SEND TO RX
            STR             /WAIT FOR REQUEST FOR SECTOR ADDRESS
            JMP.-1
            TAD (SECT       /GET SECTOR ADDRESS
            XDR             /AND TRANSFER TO RX
            CLA             /XDR DOES NOT CLEAR AC
            STR             /NOW WAIT FOR REQUEST FOR TRACK ADDRESS
            JMP.-1
            TAD (TRACK      /GET TRACK ADDRESS
            XDR             /SEND TO RX
            CLA
            SDN             /NOW WAIT FOR FUNCTION TO COMPLETE
            JMP.-1
            SER             /WERE THERE ERRORS?
            JMP GOODRD      /NO, GO ON
            JMP ERRREC      /YES, GO TO ERROR RECOVERY
GOODRD,     .
            .
            .
            .
ERRREC,     .
            .
            .
            .
```

Floppy Disk Programming Example 2

```
/EMPTY SECTOR BUFFER EXAMPLE

            TAD (BUFST-1    /SET UP BUFFER POINTER
            DCA AUTO1Ø      /LOAD INTO AUTO-INDEX REGISTER
            TAD (FUNCT      /ØØØ ØØØ ØØØ Ø1Ø (Bits <8:1Ø> = ØØ1)
            LCD             /SEND TO RX
MORDAT,     STR             /WAIT FOR DATA WORD
            JMP DONTST      /IF NOT SET, TRY DONE FLAG
            XDR             /WORD AVAILABLE
            DCA I AUTO1Ø
DONTST,     SDN             /BUFFER EMPTIED YET?
            JMP MORDAT      /NO, GO BACK AND WAIT FOR NEXT WORD
            SER             /BUFFER EMPTY, CHECK ERRORS
            JMP GOOD        /NO ERRORS, GO ON
            JMP ERRREC      /ERRORS, GO TO RECOVERY ROUTINE
GOOD,       .
            .
            .
ERRREC,     .
            .
            .
```

5.6 PROGRAMMING CONSIDERATIONS

The proper procedure for servicing interrupts in time critical applications is discussed in the following paragraph.

5.6.1 Critical Interrupt Servicing

It is possible that the VT278 will periodically disable the main memory interrupt system for periods of up to 1 millisecond during the process of updating the display buffer memory. In a time critical application, the keyboard and display may be disabled to allow the CPU to dedicate itself to interrupt servicing. The effect of disabling the keyboard and display is that the CRT controller continues to refresh the screen with the last contents of the display buffer memory and the keyboard no longer responds to keystrokes (no keyclick). Any user programs that change the status of the keyboard/display must be carefully debugged. Failure to return to the correctly enabled state requires that the system be powered down to reactivate the keyboard.

The recommended sequence to disable the keyboard and display is:

```
CLA
6125
6115
```

and to reenable:

```
CAF
CLA IAC
6125
6115
```

NOTE: When display/keyboard interrupts are disabled the following loop hangs after some number of characters have been transmitted:

```
IAC
TLS
TSF
JMP .-1
JMP .-4
```

CHAPTER 6
PROGRAMMING THE DP278

6.1 INTRODUCTION

This chapter contains general programming information and register bit assignments for the DP278 Communication Option. The DP278 is available in two variations:

1. DP278-A is the basic two-port option. Each port contains a standard communications device (SCD) whose internal registers can be programmed for asynchronous and byte-synchronous operation.

2. DP278-B is similar to DP278-A, but each port contains an additional extended synchronous device (ESD) that can be programmed to permit both byte-synchronous and bit-synchronous (BITSTUFF) operation.

6.2 DP278 INTERRUPTS

The DP278 can generate three types of interrupts, all of which may be vectored to determine the cause of interrupt.

1. Receiver interrupts
2. Transmitter interrupts
3. Modem change interrupts

6.2.1 Receiver Interrupts

Each port can generate a receiver interrupt, if enabled, when either a new character is assembled from the communications line or when a parity, overrun, or CRC (ESD only) error, is detected.

6.2.2 Transmitter Interrupts

Each port may generate a transmitter interrupt, if enabled, when either it is permissable to load a new character for transmission, or when a transmitter underflow occurs (ESD only).

6.2.3 Modem Change Interrupt

When either port detects a change of state on any incoming modem control line, if enabled, this interrupt flags the change of modem state. The user program may, at any time, issue an IOT instruction to examine the contents of either port's modem control register.

6.3 DP278-A STANDARD COMMUNICATIONS OPTION

The DP278-A standard communications option is a dual universal asynchronous/synchronous data communications controller. You can

program it to transmit and receive either synchronous or asynchronous data in full- or half-duplex operation. Transmitted data is received from the VT278 DX bus and serialized for line transmission. Received data is presented in an 8-bit parallel format to the DX bus.

6.3.1 Asynchronous Operation (SCD)

Under asynchronous operation, the SCD can be programmed to:

- Transmit/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, and
- Detect parity, framing, and overrun errors.

6.3.2 Synchronous Operation (SCD)

Under synchronous operation, the SCD can be programmed to:

- Transmit/receive 5-, 6-, 7-, or 8-bit characters
- Compute and insert a parity bit
- Recognize and synchronize upon one or two SYN characters
- Perform automatic SYN or SYN-DLE insertion and SYN or DLE character stripping, and
- Allow for automatic SYN-SYN or DLE-SYN character fill.

6.3.3 Clocking (SCD)

For the SCD, use internal clocking for both asynchronous and maintenance mode operation. Use external clocking for synchronous operation.

6.3.4 Maintenance Mode Operation (SCD)

There are two types of maintenance checks:

1. Local loopback -- causes the communications device to internally connect its transmitter output to receiver input and, therefore, check for parity, status, overrun, and framing errors.
2. Remote loopback -- permits a remote modem to send data to the communications device and have that data sent back without sending it to the DX bus. The communications device status register reports any send or receive errors.

6.3.5 Standard Communications Device (SCD) Registers

Table 6-1 lists the SCD internal registers that can be accessed under program control.

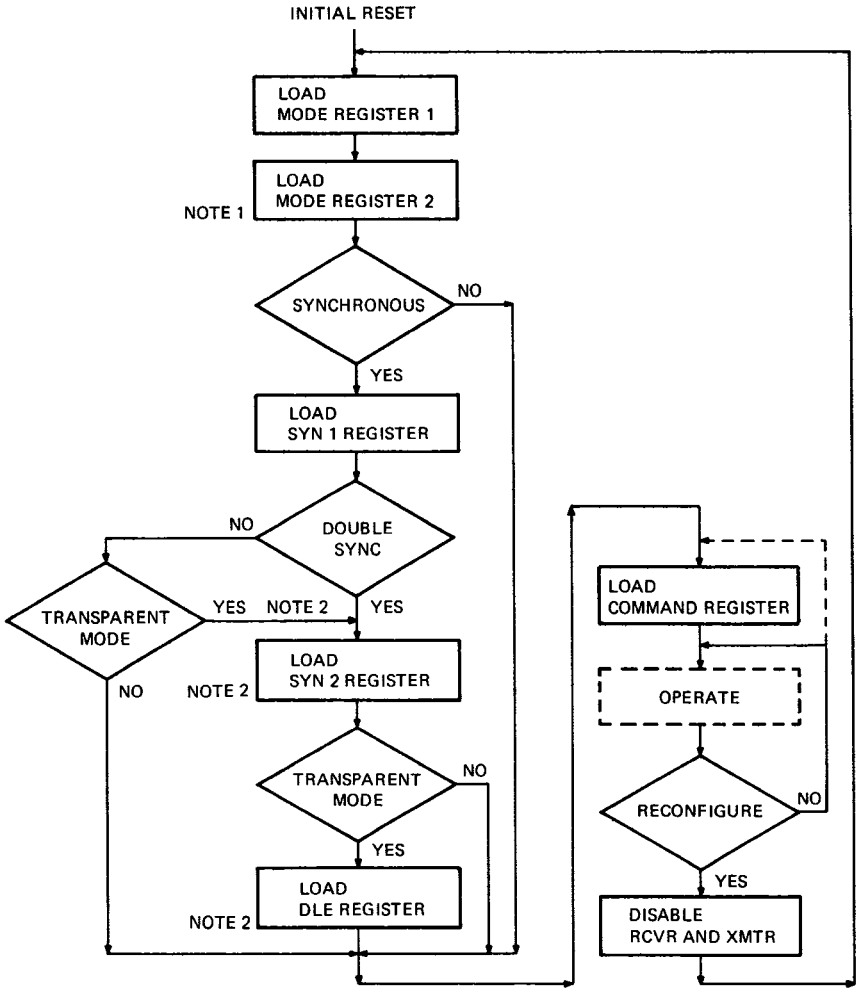
Table 6-1 SCD Internal Register Addressing

Register	Type	AC<1:3> Read	AC<1:3> Write
Receive holding register	Read	ØØØ	--
Transmit holding register	Write	--	ØØ1
SYN1, SYN2, or DLE	Write only	--	Ø11
Internal status	Read only	Ø1Ø	--
Mode register 1	Read/write	1ØØ	1Ø1
Mode register 2	Read/write	1ØØ	1Ø1
Command	Read/write	11Ø	111

Some registers are read only, some are write only, and some can be either read from or written to. Each register is accessed by setting accumulator bits AC<1:3> with the correct register addressing information and then executing the appropriate IOT instruction to read from, or write to, a particular port. The IOT instructions are for device codes 3Ø, 31, 34, and 35.

In addition, the DP278-A has a modem register, a port control register, and a port status register for each port. This allows you to program modem controls and to read port and modem status. These registers are accessed by IOT instructions for device code 36 and are described in Paragraph 6.5.2.

Programming the DP278 requires an initialization process to configure the internal registers before data transmission. Figure 6-1 demonstrates a typical programming sequence for initializing the SCD. SCD register bit definitions are described in Tables 6-2 through 6-6.

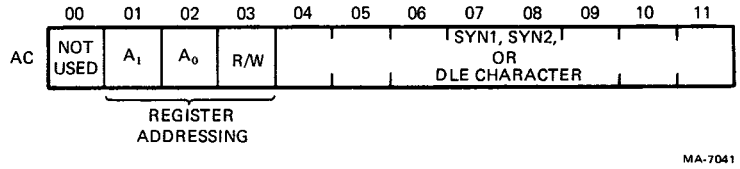


NOTES
1. MODE REGISTER 1 MUST BE WRITTEN BEFORE 2 CAN BE WRITTEN. MODE REGISTER 2 NEED NOT BE PROGRAMMED IF EXTERNAL CLOCKS ARE USED.
2. SYN1 REGISTER MUST BE WRITTEN BEFORE SYN2 CAN BE WRITTEN, AND SYN2 BEFORE DLE CAN BE WRITTEN.

MA-6444

Figure 6-1 SCD Programming Flowchart

Table 6-2 SCD: SYN1, SYN2, or DLE Register Bit Definitions

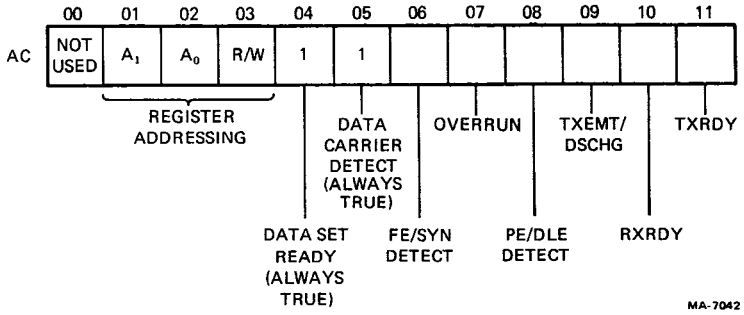


Bits	Definition
Ø	Not used
1--3	Register addressing A_1 $A_Ø$ R/W Ø 1 1 = Write SYN1, SYN2, or DLE register (see note)
4--11	SYN1, SYN2, or DLE character (bits 7--Ø)

NOTE: SYN1, SYN2, and DLE are accessed via write operations with conditions A_1=Ø, $A_Ø$=1, and R/W=1. The first operation loads SYN1; the next loads SYN2; the third loads DLE. If the required number of accesses is exceeded, the internal sequencer cycles to point at the first register. The pointer can be reset to the SYN1 register via a "read command register" operation or by an IOT 6367 (RESET communications device).

IOT instruction 6367 resets all registers in the associated devices in a port. Therefore, the registers must be reprogrammed.

Table 6-3 SCD: Internal Status Register Bit Definitions

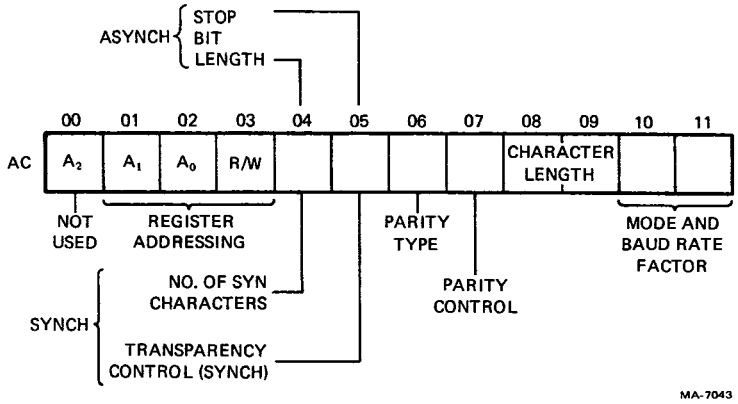


Bits	Description
Ø	Not used
1--3	Register addressing A_1 $A_Ø$ R/W Ø 1 Ø = Read status register
4	Data set ready -- always set to logical 1.
5	Data carrier detect -- always set to logical 1.
6	FE/SYN detect -- FE (framing error) detected during Asynch mode operation. Set if character received was not followed by a valid stop bit. SYN (SYN character) detected during Synch mode operation. Asynch: Ø=normal; 1=framing error. Synch: Ø=normal; 1=SYN character detected.
7	Overrun -- set during Asynch mode if preceding character was lost because receiver failed to keep up with incoming character rate on this port. Ø=normal 1=overrun error
8	PE/DLE detect -- set if character was received with a parity sense opposite to that selected for this port. For Synch operation, also may mean a DLE character was received. Asynch: Ø=normal; 1=parity error. Synch: Ø=normal; 1=parity error or DLE character received.
9	TxEMT/DSCHG -- set when transmit shift register is empty. Ø=normal 1=transmit shift register empty

Table 6-3 SCD: Internal Status Register Bit Definitions (Cont)

Bits	Description
1Ø	RxRDY Ø=receive holding register empty 1=receive holding register has data
11	TxRDY Ø=transmit holding register busy 1=transmit holding register empty

Table 6-4 SCD: Mode Register 1 Bit Definitions



Bits	Description
Ø	Unused
1--3	Register addressing A_1 $A_Ø$ R/W 1 Ø Ø = Read mode register (see note 1) 1 Ø 1 = Write mode register (see note 1)
4	For synch operation, this bit represents the number of SYN characters: Ø=double SYN 1=single SYN
5	For synch operation, this bit controls transparency: Ø=normal 1=transparent
4--5	For asynch operation, these bits represent the stop code length: ØØ=invalid Ø1=1 stop bit 1Ø=1.5 stop bit (for 5-bit characters) 11=2 stop bits.
6	Parity type -- specifies whether port is to have even or odd parity generated and checked. Has no effect unless bit 7 (parity control) is set. Ø=odd 1=even

Table 6-4 SCD: Mode Register 1 Bit Definitions (Cont)

Bits	Description
7	Parity control -- specifies whether a parity bit is generated on transmission and checked and stripped on reception: Ø=no parity bit 1=parity enabled.
8--9	Character length -- specifies the character length (not counting start bit, stop bits, and parity bit if enabled) for selected port. ØØ=5 bit chars. Ø1=6 bit chars. 1Ø=7 bit chars. 11=8 bit chars.
1Ø--11	Mode and baud rate factor -- selects mode (asynch or synch) and baud rate factor (see note 2). ØØ=synchronous 1 X rate Ø1=asynchronous 1 X rate 1Ø=asynchronous 16 X rate 11=asynchronous 64 X rate

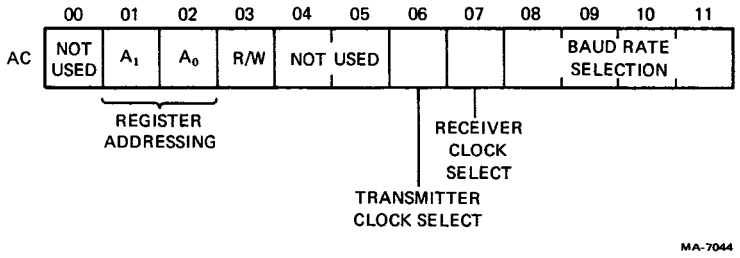
NOTES:

1. Mode registers 1 and 2 are accessed via read operations with A_1=1, $A_Ø$=Ø, and R/W=Ø, or write operations with A_1=1, $A_Ø$=Ø, R/W=1. The first read (write) operation addresses mode register 1; a second operation addresses register 2. If the required number of accesses are exceeded, the internal sequencer cycles to point at the first register. The pointer can be reset to register 1 by performing a "read command register" operation or by an IOT 6367 (RESET communications device).

   IOT instruction 6367 resets all registers in the associated devices in a port. Therefore, the registers must be reprogrammed.

2. Asynchronous baud rate factor only applies if external clock is selected for SCD mode register 2 (Table 6-4). However, one of the asynchronous baud rate factors must be chosen. When internal clock mode is selected from mode register 2, the clock output from the SCD is 16X rate.

Table 6-5 SCD: Mode Register 2 Bit Definitions



Bits	Description
Ø	Unused
1--3	Register addressing A_1 $A_Ø$ R/W 1 Ø Ø = Read mode register 2 (see note) 1 Ø 1 = Write mode register 2 (see note)
4--5	Unused
6	Transmitter clock Ø=external 1=internal
7	Receiver clock Ø = external 1 = internal
8--11	Baud rate selection -- specifies receiver and transmitter speed of operation for selected port.

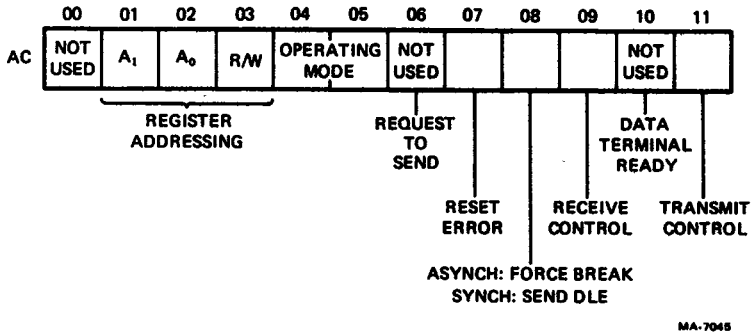
Bits 8	9	10	11	Baud
Ø	Ø	Ø	Ø	50
Ø	Ø	Ø	1	75
Ø	Ø	1	Ø	110
Ø	Ø	1	1	134.5
Ø	1	Ø	Ø	150
Ø	1	Ø	1	300
Ø	1	1	Ø	600
Ø	1	1	1	1200
1	Ø	Ø	Ø	1800
1	Ø	Ø	1	2000
1	Ø	1	Ø	2400
1	Ø	1	1	3600
1	1	Ø	Ø	4800
1	1	Ø	1	7200
1	1	1	Ø	9600
1	1	1	1	19200

Table 6-5 SCD: Mode Register 2 Bit Definitions (Cont)

NOTE: Mode registers 1 and 2 are accessed via read operations with A_1=1, $A_\emptyset$=Ø, and R/W=Ø, or write operations with A_1=1, $A_\emptyset$=Ø, R/W=1. The first read (write) operation addresses mode register 1; a second operation addresses register 2. If the required number of accesses is exceeded, the internal sequencer cycles to point at the first register. The pointer can be reset to register 1 by performing a "read command register" operation or by an IOT 6367 (RESET communications device).

IOT instruction 6367 resets all registers in the associated devices in a port. Therefore, the registers must be reprogrammed.

Table 6-6 SCD: Command Register Bit Definitions



Bits	Definition
Ø	Not used
1--3	Register addressing A_1 $A_Ø$ R/W 1 1 Ø = Read command register 1 1 1 = Write command register
4--5	Operating mode ØØ=normal operation Ø1=Asynch: automatic echo mode (echoplex); Synch: SYN and/or DLE stripping mode. 1Ø=local loopback 11=remote loopback
6	Request to send (RTS)--unused Ø=force RTS output high 1=force RTS output low
7	Reset error Ø=normal 1=reset error flag in status register (framing error, overrun, PE/DLE detect)
8	Force break/send DLE Asynch: force break Ø=normal 1=force break. Synch: send DLE Ø=normal 1=send DLE.

Table 6-6 SCD: Command Register Bit Definitions (Cont)

Bits	Definition
9	Receive Control Ø=disable 1=enable NOTE: Whenever you enable the SCD Receive Control, you must also perform a "dummy read." This operation clears the holding register before any incoming characters are received, to ensure the device will operate properly. You must perform the following steps: 1. Set AC<1:3> and AC9 to 1, and all other desired bits from Table 6-6, to write the Receive Control bit to the command register. 2. Execute an IOT 6316 (port Ø) or IOT 6356 (port 1). 3. Set AC<1:3> to Ø to read the receive holding register (Table 6-1). 4. Execute an IOT 6316 (port Ø) or IOT 6356 (port 1).
1Ø	Data Terminal Ready (DTR)--unused Ø=force DTR output high 1=force DTR output low
11	Transmit Control Ø=disable 1=enable

6.4 DP278-B EXTENDED COMMUNICATION OPTION

This version of the DP278 communication option adds the capability of bit synchronous communication (BITSTUFF) for such protocols as X.25, HDLC, and SDLC. You can select, under program control, whether a particular port operates as an SCD or ESD communication device. Refer to Paragraph 6.3 for SCD programming information.

6.4.1 Synchronous Operation (ESD)

The DP278-B communication option can be programmed to:

- Operate in a bit-oriented protocol (BOP) or a byte-control protocol (BCP);
- Select character length (1 to 8 bits for BOP and 5 to 8 bits for BCP);
- Provide SYNC or secondary address comparison for BCP/BOP;
- Automatically detect and generate special BOP control sequences, such as FLAG, ABORT, and GA;

- Perform SYNC generation, detection, and stripping for BCP;
- Perform CRC generation and checking.

6.4.2 Clocking (ESD)

All clocking information for the ESD is derived from the incoming modem lines. However, in maintenance mode, the SLU1 clock is substituted for the transmitter clock (TXCLK) input to each ESD.

6.4.3 Maintenance Mode Operation (ESD)

When maintenance mode is selected for the ESD, a local loopback check similar to the SCD is enabled. Local loopback causes the communication device to internally connect its transmitter output to its receiver input.

6.4.4 Extended Synchronous Device (ESD) Registers

The following ESD internal registers can be accessed under program control:

Register	Type
Receiver data buffer	Read only
Receiver status	Read only
Transmitter data buffer	Write only
Transmitter status	Read/write
Parameter control and control sync/address (PCSAR)	Read/write*
Data length select (PCSCR)	Read/write

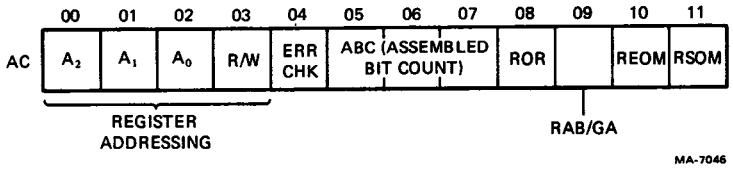
* Two registers; each R/W.

Some registers are read only, some are write only, and some can be either read from or written to. Each register is accessed by setting accumulator bits AC<Ø:3> with the correct register addressing information. Then the appropriate IOT instruction is executed to read from, or write to, a particular port. IOT instructions are for device codes 3Ø, 31, 34, and 35.

In addition, the communication option has a modem register, a port control register, and a port status register for each port. This allows you to program modem control and to read port and modem status. These registers are accessed by IOT instructions for device code 36 and are described in Paragraph 6.5.2.

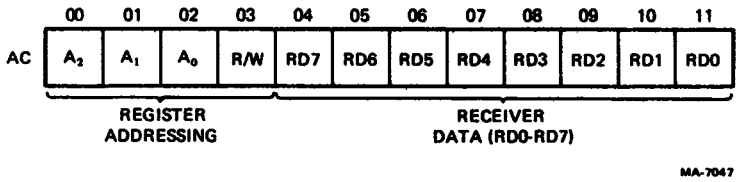
ESD register bit definitions are described in Tables 6-7 through 6-13.

Table 6-7 ESD: Receiver Data/Status Register (RDSR) Receiver Status Bit Definitions



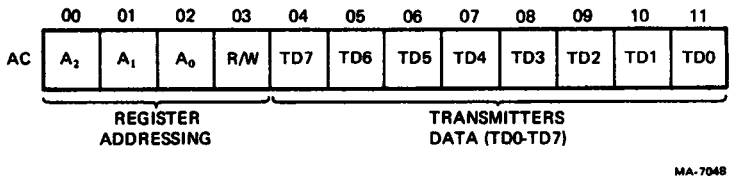
Bits	Definition
Ø--3	Register addressing A_2 A_1 $A_Ø$ R/W Ø Ø 1 Ø = Read receiver status register
4	ERR CHK (error check) -- in BOP (bit oriented protocols), this bit is set if CRC is selected and received in error. Check this bit when bit REOM=1. In BCP (byte control protocols) 1) This bit is set if parity is selected and received in error. 2) If CRC is selected it is tested at end of each byte. ERR CHK=1 if CRC is good; ERR CHK=Ø if CRC not good. Controller must determine last byte of message.
5--7	ABC (assembled bit count) -- in BOP mode only, examine when REOM=1. ABC=Ø, message terminated on stated boundary. ABC=XXX, message terminated by (FLAG or GA) on unstated boundary. Binary value of ABC=number of valid bits available in receiver data buffer (RDB) right-justified.
8	ROR (receiver overrun) -- set to 1 when received data transferred into RDB and previous data was not read, indicating failure to service RDA (receiver data available) within one character time. Cleared when receiver status register is read, or when RXENA is dropped.
9	RAB/GA (receiver abort or go ahead) -- in BOP mode only, if LM (loop mode)=Ø, this bit is set if an Abort character is received. If LM=1, this bit is set if a Go Ahead character is received. Bit is cleared when receiver status register is read or when RXENA is dropped.
1Ø	REOM (receiver end of message) -- in BOP mode only, this bit is set to 1 when the last byte of data is loaded into the RDB (receiver data buffer) or when an Abort character is received. Bit is cleared when receiver status register is read or when RXENA is dropped.
11	RSOM (receiver start of message) -- in BOP mode only, this bit is set to 1 when first non-flag (address byte) is loaded into RDB (receiver data buffer). Bit is cleared when second byte is loaded into RDB.

Table 6-8 ESD: Receiver Data/Status Register (RDSR), Receiver Data Buffer Bit Definitions



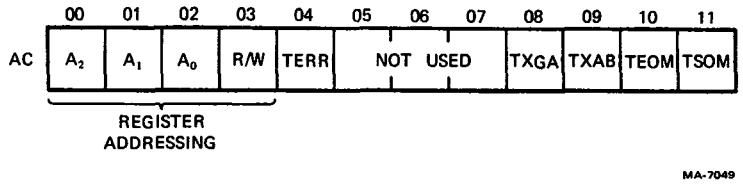
Bits	Definition
Ø--3	Register addressing A_2 A_1 $A_Ø$ R/W Ø Ø Ø Ø = Read receiver data buffer
4--11	Receiver data (bits RD<7:Ø>)

Table 6-9 ESD: Transmitter Data/Status Register (TDSR) Transmitter Buffer Bit Definitions



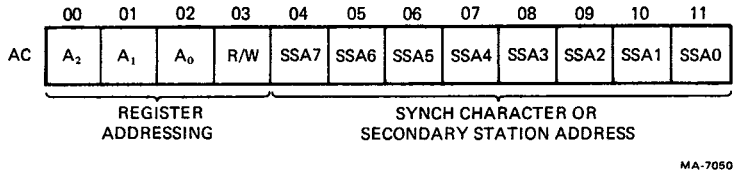
Bits	Definition
Ø--3	Register addressing A_2 A_1 $A_Ø$ R/W Ø 1 Ø 1 = Write transmit data buffer
4--11	Transmitter data (bits TD<7:Ø>)

Table 6-10 ESD: Transmitter Data/Status Register (TDSR) Transmitter Status and Control Register Bit Definitions



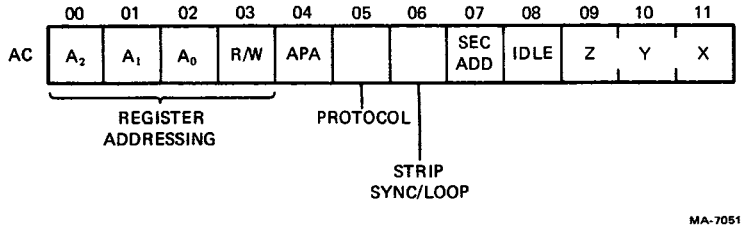
Bits	Definition
0--3	Register addressing A_2 A_1 A_0 R/W 0 1 1 0 = Read transmitter status register 0 1 1 1 = Write transmitter control register
4	TERR (transmitter error) -- read only bit. Underflow, set high when TDB not loaded in time to maintain continuous transmission. In BOP, automatically transmit: Abort if IDLE=0; Flag if IDLE=1. In BCP, automatically transmit: Sync if IDLE=0; Mark, cleared by TSOM if IDLE=1.
5--7	Not used
8	TXGA (transmit go ahead) -- R/W bit. In BOP mode only, modifies character called for by TEOM. If set GA character sent in place of Flag.
9	TXAB (transmitter abort) -- R/W bit. In BOP mode only, TXAB=1: finish present character, then: transmit Abort if IDLE=0; transmit Flag if IDLE=1.
10	TEOM (transmit end of message) -- R/W bit. Used to terminate a message. In BOP mode: TEOM =1; send CRC, then Flag; if TXENA=1 and TEOM=1, continue to send Flag; if TXENA=0 and TEOM=1, Mark line. In BCP: IDLE=0, TEOM=1: send CRC, then SYNCH IDLE=1, TEOM=1: send CRC, then character from the transmitter data buffer. If CRC not selected, do not set TEOM.
11	TSOM (transmit start of message) -- R/W bit. If TXENA=1, TSOM initiates start of message. In BOP mode: TSOM=1 generates Flag and continues to send Flags until TSOM=0, then begin data. In BCP mode: IDLE=0, send SYNCH. IDLE=1 transmit out of TDB. In BOP mode there is also a special space sequence of 16 zeros initiated by TSOM=1 and TEOM=1. Special space sequence is followed by Flag. CRC generation (if selected) begins when TSOM=0.

Table 6-11 ESD: Parameter Control Synch/Address Register (PCSAR) SYNCH/Secondary Address Register Bit Definitions



Bit	Definition
Ø--3	Register addressing A_2 A_1 $A_Ø$ R/W 1 Ø Ø Ø = Read Synch/address register 1 Ø Ø 1 = Write Synch/address register
4--11	Synch character/secondary status address -- in BCP (byte control protocol) mode, bits 4--11 contain the Synch character. Bit 11 is the least significant bit. In BOP (bit oriented protocols) mode, and when operating in secondary address mode, these bits contain the desired secondary station address.

Table 6-12 ESD: Parameter Control Synch/Address Register (PCSAR) or Mode Control Register Bit Definitions



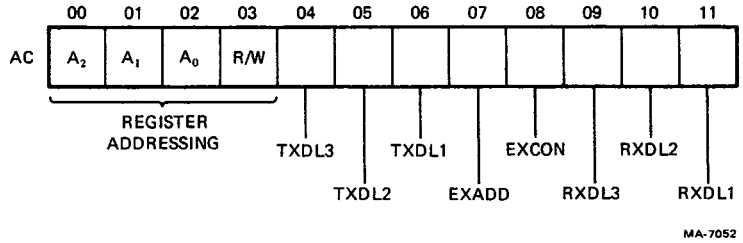
Bits	Definition
Ø--3	Register addressing A_2 A_1 $A_Ø$ R/W 1 Ø 1 Ø = Read mode control register 1 Ø 1 1 = Write mode control register
4	APA (all parities address) -- if selected, modifies secondary mode so that secondary address of 8 ones activates the RDP (receive data path).
5	Protocol -- Ø=BOP (bit oriented protocol) -- ADCCP, X.25, HDLC, SDLC. 1=BCP (byte control protocol) -- DDCMP, BISYNC.
6	Strip/Sync loop -- strip sync or loop mode. Effects receiver only. In BOP mode allows recognition of a GA (go ahead) character. In BCP mode, after second Sync, strip Sync. When first data character detected, set RXACT=1, stop stripping sync patterns.
7	SEC ADD (secondary address mode) -- in BOP mode only, looks for address match prior to activating receiver. When transmitting, the secondary address must be loaded into the TDB.
8	IDLE (Idle mode select) -- affects transmitter only. In BOP mode, controls type of character sent when TSAB (transmit abort) asserted or in event of data underflow. In BCP mode, controls method of initial Sync character transmission and underflow.
9--11	XYZ* (error control bits) -- Z Y X Ø Ø Ø $X^{16} + X^{12} + X^5 + 1$ CCITT -- initialize to "1" Ø Ø 1 $X^{16} + X^{12} + X^5 + 1$ CCITT -- initialize to "Ø" Ø 1 Ø Not used Ø 1 1 $X^{16} + X^{12} + X^2 + 1$ CRC16

Table 6-12 ESD: Parameter Control Synch/Address Register (PCSAR) or Mode Control Register Bit Definitions (Cont)

Bits	Definition
	1 Ø Ø odd parity, CCP mode only
	1 Ø 1 even parity, CCP mode only
	1 1 Ø Not used
	1 1 1 Inhibit all error detection

* Do not modify XYZ until both transmitter and receiver are idle.

Table 6-13 ESD: Data Length Select Register (PCSCR) Bit Definitions



Bits	Definition
0--3	Register addressing A_2 A_1 A_0 R/W 1 1 1 0 = Read data length select register 1 1 1 1 = Write data length select register
4--6	TDXL (transmitter data length) **TXDL3 (Bit 4) / TXDL2 (Bit 5) / TXDL1 (Bit 6) / Length (Bits/Character)** 0 0 0 8 1 1 1 7 1 1 0 6 1 0 1 5 1 0 0 4* 0 1 1 3* 0 1 0 2* 0 0 1 1*
7	EXADD (extended address field) -- in receiver only, LSB of address byte tested for a "1." If no, continue receiving address bytes, if yes, go into control field.
8	EXCON (extended control field) -- in receiver only, if set, receives control field as two 8-bit bytes.

Table 6-13 ESD: Data Length Select Register (PCSCR) Bit Definitions (Cont)

Bits	Definition			
9--11	RXDL (receiver data length)			
	RXDL3 (Bit 9)	RXDL2 (Bit 10)	RXDL1 (Bit 11)	Length (Bits/Characters)
	0	0	0	8
	1	1	1	7
	1	1	0	6
	1	0	1	5
	1	0	0	4
	0	1	1	3
	0	1	0	2
	0	0	1	1

* Only use these character lengths in BOP mode operation.

6.5 DP278 IOT INSTRUCTIONS

This section describes the IOT instructions that control the internal DP278 registers and the IOT instructions that provide modem control and port status operations.

A 6XY4 instruction performs the following functions (where XY may be 30, 31, 34, or 35, and not 36):

1. For a read device, data is received from the device, ORed with data in the AC, and the result is stored in the AC.

2. For a write device, the contents of the AC are sent to the device.

A 6XY6 IOT instruction performs the following functions (where XY may be 30, 31, 34, or 35, and not 36):

1. For a read device, data is received from the device and loaded in the AC.

2. For a write device, the contents of the AC are sent to the device; then the AC is cleared.

6.5.1 Internal Register Instructions

The IOT instructions in Table 6-14 control the DP278 internal registers. Before using instructions 6306, 6316, 6346, or 6356, set the accumulator to the bit assignments in Tables 6-2 through 6-13.

Table 6-14 DP278 Internal Register IOT Instructions

Mnemonic	Octal Code	Function
SPRFØ	63ØØ	Set port Ø receive flag
KPRFØ	63Ø1	Skip on port Ø receive flag and clear it
	63Ø2	Clear AC
PRIEØ	63Ø5	Load port Ø receive interrupt enable
PRBØ	63Ø6	Read port Ø comm. device
SPTFØ	631Ø	Set port Ø transmit flag
KPTFØ	6311	Skip on port Ø transmit flag and clear it
	6312	NOP
PTIEØ	6315	Load port Ø transmit interrupt enable
PTBØ	6316	Write port Ø comm. device
SPRF1	634Ø	Set port 1 receive flag
KPRF1	6341	Skip on port 1 receive flag and clear it
	6342	Clear AC
PRIE1	6345	Load port 1 receive interrupt enable
PRB1	6346	Read port 1 comm. device
SPTF1	635Ø	Set port 1 transmit flag
KPTF1	6351	Skip on port 1 transmit flag and clear it
	6352	NOP
PTIE1	6355	Load port 1 transmit interrupt enable
PTB1	6356	Write port 1 comm. device

6.5.2 Port Control, Port Status, and Modem Register Instructions

The IOT instructions described Table 6-15 are for modem control and port status operations.

6.6 DOWN-LINE LOADING THE VT278

When the VT278 is equipped with a DP278, programs can be down-line loaded from a host computer to the VT278 by following the operational procedures for terminal mode I or terminal mode II operation described in Chapter 3. Programming considerations that must be made regarding the protocol for transmission and the format of information to be transmitted is discussed in the following paragraphs.

NOTE: If the down-line load process has been successful, all control of the VT278 is turned over to, and becomes the full responsibility of, the application program which has been loaded. Digital Equipment Corporation takes no further responsibility for the control of any modems or telephone equipment connected to the VT278 at this point.

Table 6-15 DP278 Port Control, Port Status, and Modem Register IOT Instructions

Mnemonic	Octal Code	Function
SMF	6360	Set modem change flag
KMF	6361	Skip on modem change flag and clear it
PCRØ	6362	Write Port Ø Control Register (does not clear AC after transfer) Before executing this IOT, set AC as follows: AC <Ø:3> — Unused 4 — Port Ø terminal timing enable 5 — Port Ø ESD transmitter enable 6 — Port Ø ESD receiver enable 7 — Port Ø request to send 8 — Port Ø terminal ready 9 — Port Ø signalling rate 1Ø — Port Ø ESD maintenance select 11 — Port Ø device select (Ø=SCD; 1=ESD) If bit 11 changes the selected device, this instruction must be followed by a 6367.
PSR	6363	Read Port X Status Register (where X=Ø or 1). Clears AC before transfer. Before executing this IOT, set ACØ for port Ø (ACØ=Ø) or port 1 (ACØ=1). Status bits in AC, after 6363 instruction is executed are: Ø — Port select (Ø=port Ø, 1=port 1) 1 — Unused (AC<1>=Ø) 2 — Port X ESD transmitter active (1=true) 3 — Port X ESD receiver active (1=true) 4 — Port X ESD transmitter buffer empty (1=true) 5 — Port X SCD transmitter buffer empty (Ø=true) 6 — Port X ESD receiver data available (1=true) 7 — Port X SCD receiver data available (Ø=true) 8 — Port X ESD transmitter status available (1=true) 9 — Port X ESD receiver status available (1=true)

Table 6-15 DP278 Port Control, Port Status, and Modem Register IOT Instructions (Cont)

Mnemonic	Octal Code	Function
		10 Port X device select (0=SCD, 1=ESD) 11 Port X ESD present* (0=ESD present)
PMR	6364	Read Port X Modem Register (clears AC before transfer). Before executing this IOT, set AC0 for port 0 (AC0=0) or port 1 (AC0=1). Modem register bits read into AC after executing 6364 are: **AC — Function** 0 — Port select (0=port 0, 1=port 1) <1:3> — Unused (AC<1:3>=0) 4 — Port X request to send 5 — Port X terminal ready 6 — Port X signalling rate 7 — Port X signal quality 8 — Port X incoming call 9 — Port X receiver ready 10 — Port X data mode 11 — Port X clear to send
MCIE	6365	Load Modem Control Interrupt Enable before executing this IOT; set AC as follows: **AC — Function** <0:10> — Ignored 11 — Modem control interrupt enable (1=enabled)
PCR1	6366	Write Port 1 Control Register (does not clear AC after transfer). Before executing this IOT, set AC as follows: **AC — Function** <0:3> — Unused 4 — Port 1 terminal timing enable 5 — Port 1 ESD transmitter enable 6 — Port 1 ESD receiver enable 7 — Port 1 request to send 8 — Port 1 terminal ready 9 — Port 1 signalling rate 10 — Port 1 ESD maintenance select 11 — Port 1 device select (0=SCD, 1=ESD)

Table 6-15 DP278 Port Control, Port Status, and Modem Register IOT Instructions (Cont)

Mnemonic	Octal Code	Function
		If bit 11 changes the selected device, this instruction must be followed by a 6367.
RACD	6367†	Reset active communications device. Set AC0 to selected port before executing this IOT. AC0=0 (port 0) AC0=1 (port 1) AC<1:11> unused Use this IOT whenever the active device in that port is changed via a 6362 or 6366 instruction.

* If ESD chip is not present in a port, all associated ESD signals are set to "1."

† IOT instruction 6367 resets all internal registers in the associated devices in a port. Therefore, the registers must be reprogrammed.

6.6.1 Down-Line Loading Protocol

The protocol for down-line loading consists of the following steps:

1. The host computer sends the sequence **ESC L** (033_8 114_8) to the VT278. The VT278 could initiate this sequence by transmitting a character sequence to the host to cause the host's transmission of the **ESC L** sequence.

2. The DP278 firmware recognizes the **ESC L** sequence and starts execution of its down-line load routine.

3. The host computer sends characters to the DP278 using the encoded format described in Paragraph 6.6.2. The encoding may be done independent of transmission. That is, a host computer utility may encode application programs, and store them for future transmission.

4. The DP278 firmware receives and decodes the characters.

5. The DP278 sends an ESCape sequence to the host when the load is completed:
   ESC G (033_8 107_8) = successful load
   ESC B (033_8 102_8) = unsuccessful load.

6. If the down-line load is successful, the VT278 begins executing the loaded program from location 200_8 in field 0.

6.6.2 Down-Line Load Encoding Format

For the DP278 firmware to recognize the characters sent by the host, the characters must be encoded by the host before transmission. The down-line load format is similar to the binary loader format generated by the PDP-8 assembler. Transmission must include the following information codes in the order shown:

Leader
Origin
Data
Checksum
Trailer.

The principal difference between the binary and down-line load formats is that character information is only included in the four least significant bits of the down-line load 8-bit byte. In addition, bits 1 and 2 must always be different. The result is that transmitted characters are always in the range 40_8 to 137_8, which excludes control codes.

Tables 6-16 and 6-17 summarize the differences between the two formats and describe the bit definitions required to identify each type of information transmitted. Notice that origin, data, and checksum each require three consecutive bytes for down-line load transmission, compared to only two for the binary loader.

Table 6-16 Comparison of Binary and Down-Line Load Formats

Binary Type	Binary Loader Bit Ø1 234 567	Down-Line Load* Bit Ø1 234 567
Header-trailer	1Ø ØØØ ØØØ	XØ 1ØØ ØØØ
Field	11 FFF ØØØ	XØ 11F FFØ
Origin	Ø1 JKL MNP	X1 Ø1J KLM
	ØØ RST UVW	X1 ØØN PRS
		X1 ØØT UVW
Data/checksum	ØØ JKL MNP	X1 ØØJ KLM
	ØØ RST UVW	X1 ØØN PRS
		X1 ØØT UVW

* Bits 1 and 2 are always different from each other.

Table 6-17 Binary and Down-Line Load Bit Definitions

Information Code Type	Binary Loader Bits Ø and 1	Down-Line Load Bits 2 and 3
Leader/trailer	Bit Ø=1	Bit 2=1
	1=Ø	3=Ø
Origin	Ø=Ø	2=Ø
	1=1	3=1
Data	Ø=Ø	2=Ø
	1=Ø	3=Ø
Field	Ø=1	2=1
	1=1	3=1

6.6.2.1 Leader/Trailer -- When the firmware detects bit 2=1 and bit 3=Ø, this byte is interpreted as the leader or trailer code. The down-line load transmission must begin with a leader code and end with a trailer code. When the DP278 firmware detects the trailer code, it expects the previously constructed 12-bit word to be the checksum.

For example:

```
Bits Ø1 234 567
     ØØ 1ØØ ØØØ  (transmitted byte)
```

results in the firmware interpreting this byte as either leader or trailer.

6.6.2.2 Origin -- When the firmware recognizes bit 2=Ø and bit 3=1, bits 4--7 of this byte and bits 4--7 of the next two bytes are interpreted as the origin. The firmware constructs these bits into a 12-bit word and loads the succeeding data words into consecutive addresses starting from the defined origin.

The DP278 firmware requires that all down-line loaded programs must start execution at location 200_8, field Ø. However, portions of the program may be located in other memory fields. Succeeding field and origin information will instruct the firmware to load the data that follows into newly specified fields starting from the indicated address.

For example:

```
Bits Ø1 234 567
     Ø1 Ø1Ø ØØØ  (3 consecutive transmitted bytes)
     Ø1 ØØ1 ØØØ
     Ø1 ØØØ ØØØ
```

results in address octal 2ØØ used as an origin address.

6.6.2.3 Field -- The firmware is initialized to start the down-line loaded program at field Ø, address 200_8. However, the down-line loaded program may include additional field and origin information, as discussed in Paragraph 6.6.2.2. When the firmware recognizes bit 2=1 and bit 3=1, it interprets bits 4--6 as the new field address. The current page address is unaffected.

For example:

Bits Ø1	234	567	
ØØ	11Ø	11Ø	(transmitted byte)

results in the current field address being set to field 3.

6.6.2.4 Data/Checksum -- When the firmware detects bit 2=Ø and bit 3=Ø, bits 4--7 of this byte and bits 4--7 of the next two bytes are interpreted as data. The firmware constructs a 12-bit word and stores the word in the next memory location of the current field. Bits 4--7 of the first byte are the most significant four bits, and bits 4--7 of the third byte are the least significant four bits.

For example:

Bits Ø1	234	567	
Ø1	ØØ1	111	(3 consecutive transmitted bytes)
Ø1	ØØ1	11Ø	
Ø1	ØØ1	11Ø	

results in 7756_8 being stored at the current field/page address.

If the next word constructed by the firmware is the trailer code, this data word is interpreted as the checksum. The checksum is computed by summing bits 4--7 in all transmitted bytes except for leader/trailer and field bytes. The carry beyond 12 bits is ignored.

7.1 INTRODUCTION

This chapter describes the RL278 disk controller registers and commands and the instruction set used to start the transfer of data to and from an RLØ2 disk drive.

The flowchart in Figure 7-1 shows the basic sequence of events that occur during execution of all controller commands. After the necessary registers are loaded, the command is decoded and the function control is started. Controller logic and error bits are cleared. The command function to be performed is then executed.

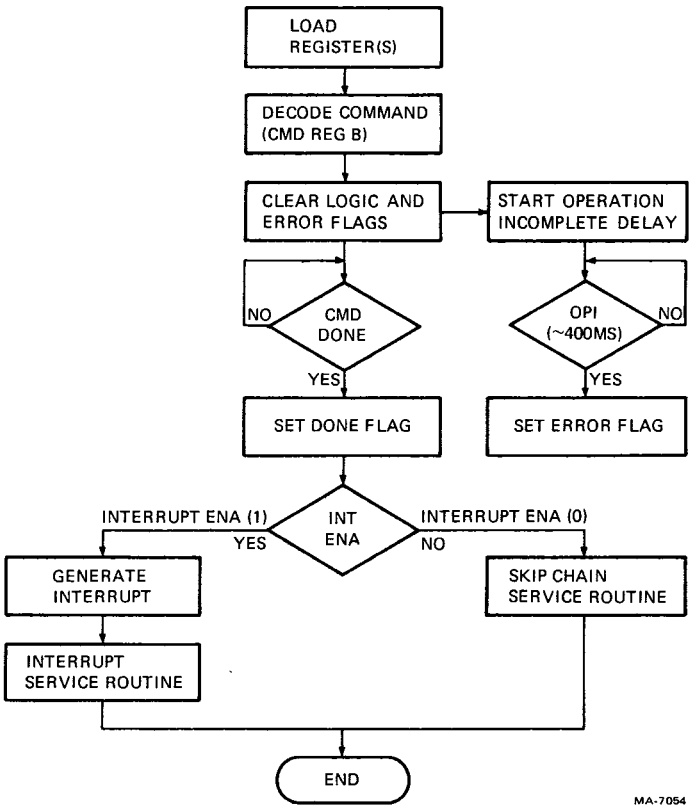


Figure 7-1 Basic Controller Operation Flowchart

When the command function is completed, the done flag and the 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) to location Ø into an interrupt service routine request before command termination.

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

Before reading or writing data, the drive heads must be positioned properly on the disk. This is done by first executing a read header command to determine where the heads are located and if the upper or lower head has been selected. If positioned at the desired track address, the read/write command can be executed directly. Otherwise, a software calculation must be performed to determine the cylinder distance the heads must move to the new location. This cylinder-distance information is then transmitted to the selected drive via a seek command.

7.2 RL278 INSTRUCTION SET

Input/output transfer (IOT) instructions are used to start disk drive(s) operation and to transfer data. Figure 7-2 shows the IOT instruction format. Bits Ø--2 contain the opcode, which must be a 6 to specify an IOT instruction. Bits 3--8 contain the device selection code and bits 8--11 contain the function code. The instructions are loaded onto the DX bus and decoded by the RL278 controller. Transfers from the AC to the controller clear the AC after the transfer is complete. Transfers to the AC from the controller clear the AC first and then the transfer takes place.

The skip instructions in the RL278 instruction set perform the skip and then clear the flag. This means that if a given flag is true, it will be cleared upon completion of the skip IOT. Table 7-1 lists and defines the instruction set for the RL278.

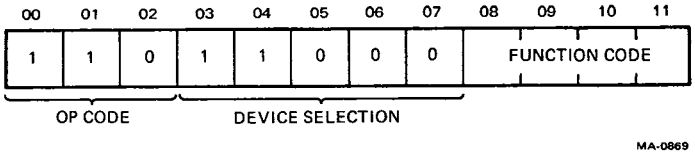


Figure 7-2 RL278 Instruction Set Format

Table 7-1 RL278 Instruction Set

Octal Code	Mnemonic	Function
6600	RLDC	Clear controller, all registers, AC and flags. (Do not use to terminate a disk function.)
6601	RLSD	Skip on function done. Then clear if set to a one.
6602	RLMA	Load break MA register from AC<0:11>
6603	RLCA	Load command register A from AC<0:11>
6604	RLCB	Load command register B from AC<0:11>, execute command
6605	RLSA	Load sector address from AC<0:5>
6606	--	Spare (will clear the AC)
6607	RLWC	Load word count from AC<0:11>
6610	RRER	Read error register into AC<0:2, 10:11>
6611	RRWC	Read word count into AC<0:11>
6612	RRCA	Read command register A into AC<0:11>
6613	RRCB	Read command register B into AC<0:11>
6614	RRSA	Read sector address into AC<0:5>
6615	RRSI	Read silo word (8-bit) into AC<4:11>
6616	--	Spare (does not clear AC)
6617	RLSE	Skip on composite error, then clear if set to a one.

7.3 ADDRESSABLE REGISTERS

The RL278 has six addressable registers that allow the CPU to communicate with the controller using IOT commands in the form of 6XXX (Table 7-1). The instructions control and monitor the controller and do not transfer data. These addressable registers are explained in more detail as follows.

1. Command register A -- contains the desired cylinder address (or difference for a seek) from which data is to be transferred, for head selection, and for a read command.

2. Command register B -- contains the function and format control for the mode (8- or 12-bit), drive select, interrupt enable, extended memory bits, and the desired function to be performed for a seek cylinder difference, head selection, and difference commands.

3. Break memory address -- contains the first memory location to/from which data will be transferred.

4. Sector address register -- contains the desired address of the first sector to be transferred.

5. Word count register -- contains the number of data words to be transferred to either memory or the disk.

6. Error register -- contains the indications for five different error conditions.

7.3.1 Command Register A

Command register A is a 12-bit addressable register that contains the cylinder address difference, head select, and direction for seek functions (Table 7-2). This register is also used during read/write operations (holds CYL ADD and HD SEL) when searching for the correct disk address.

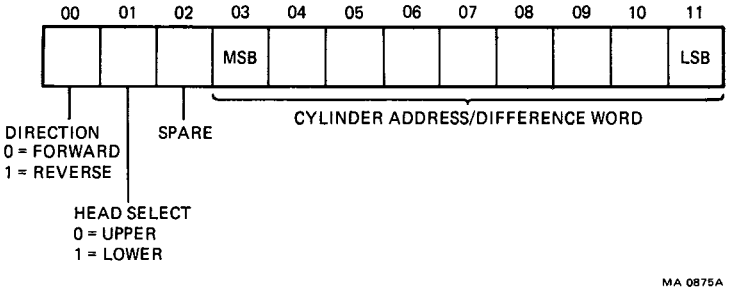
7.3.2 Command Register B

Command register B is a 12-bit addressable register that contains the mode, drive number, interrupt enable, and the function code (Table 7-3).

7.3.3 Silo Buffer After Get Status

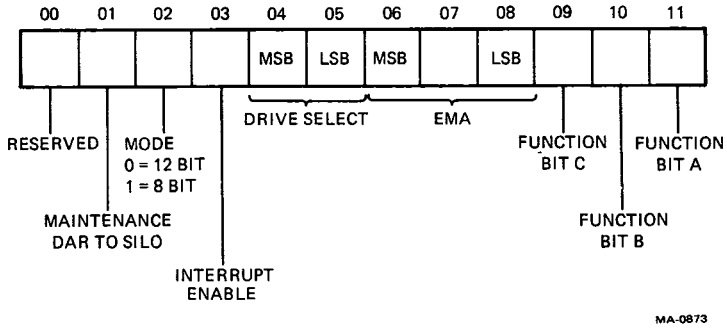
When a get status command is executed, the disk drive returns a status word to the controller. The contents of the silo buffer is divided into two words: word 1 and word 2 messages (Table 7-4). The fault LED on the drive(s) will be asserted if any bit is set in word 2.

Table 7-2 Command Register A Bit Definitions



Bit	Name	Function
AC0	Direction (DIR)	Indicates the direction in which a seek is to take place. When the bit is set, the heads move toward the spindle (to a higher cylinder address). When the bit is cleared, the heads move away from the spindle (to a lower cylinder address). The actual distance moved depends on the cylinder address difference (bits 3--11).
AC1	Head Select (HS)	Indicates which head (disk surface) is to be selected: set=lower (HD 1); clear=upper (HD0).
AC2	--	Spare
AC<3:11>	Cylinder address difference DF<3:11>	Indicates number of cylinders heads are to move on a seek and a read/write operation. Range is $0000--0777_8$ ($0--511_{10}$).

Table 7-3 Command Register B Bit Definitions

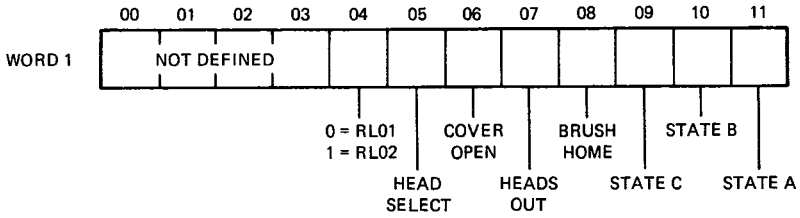


Bit	Name	Function
AC0	--	Reserved, normally cleared
AC1	Maintenance	Reads disk address register (DAR) and loads into silo for maintenance purposes. Bits 1 and 2 of command register B must be set to 1 for function to work correctly. DAR is loaded by using IOT RLCA to load command register A (for cylinder address portion) and RLCB to load the function along with the loopback bit (1) and the mode bit (2). When the function done flag sets to a 1, the silo should be read twice to yield two 8-bit words which are the DAR. The first word has the following content. (AC<0:3> are undefined for both words.) AC04 -- Cylinder add. (LSB) AC05 -- Head select AC06 -- 0 AC07 -- Head select for seek AC08 -- 1 for reset, otherwise 0 AC09 -- Direction bit AC10 -- 1 for get status, otherwise 0 AC11 -- Marker bit (always 1) The second word has the following content: AC04 -- Cylinder add. (MSB, RL02) AC05 -- Cylinder add. (MSB, RL01) AC06 -- Cylinder add. AC07 -- Cylinder add.

Table 7-3 Command Register B Bit Definitions (Cont)

Bit	Name	Function
		AC08 -- Cylinder add. AC09 -- Cylinder add. AC10 -- Cylinder add. AC11 -- Cylinder add.
AC2	Mode	When set, indicates that data field will be 256 8-bit words per sector. When 0, data field will be truncated to 170 12-bit words per sector (12-bit mode). The OS/8 operating system limits it to 128. This bit must be set when a get status or read header command is to be executed.
AC3	Interrupt enable (IE)	When this bit is set, controller is allowed to interrupt processor at end of normal command or error termination.
AC<4:5>	Drive select	Determines which drive will communicate with controller via drive bus.
AC<6:8>	Extended memory address (EMA)	Defines memory field location. Allows 32,768 locations to be addressed for processors having more than 4K of memory.
AC<9:11>	Function code	Indicates command to be executed by controller/disk subsystem. Bits 9 10 11 Command 0 0 0 Maintenance 0 0 1 Reset 0 1 0 Get status 0 1 1 Seek 1 0 0 Read header 1 0 1 Write data 1 1 0 Read data 1 1 1 Read data w/o header check

Table 7-4 Silo Buffer Bit Definitions

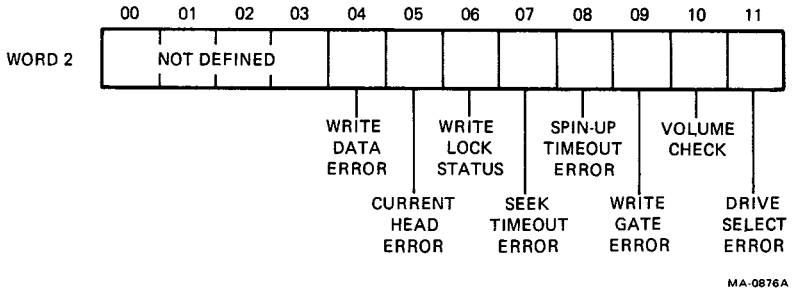


Word No. 1

Bit	Name	Function
AC<Ø:3>	--	Undefined
AC4	Drive identification	Ø=RLØ1; 1=RLØ2
AC5	Head select (HS)	Indicates currently selected head.
AC6	Cover open (CO)	Asserted when top cover is open or dust cover not in place.
AC7	Heads out (HO)	Asserted when heads are over disk. Cleared when heads retract.
AC8	Brush home (BH)	Asserted when brushes retract.
AC<9:11>	State bits	Defines state of disk drive.

Bits 9	10	11	Definition
Ø	Ø	Ø	Load state
Ø	Ø	1	Spin-up
Ø	1	Ø	Load heads
Ø	1	1	Brush cycle
1	Ø	Ø	Seek-track counting
1	Ø	1	Seek linear mode
1	1	Ø	Unload heads
1	1	1	Spin-down

Table 7-4 Silo Buffer Bit Definitions (Cont)



Word No. 2

Bit	Name	Function
AC<Ø:3>	--	Undefined
AC4	Write data error (WDE)	When asserted indicates that write gate was on but no transitions were detected on write data line.
AC5	Current head error (CHE)	Write current was detected in heads when write gate was not asserted.
AC6	Write lock (WL)	Write lock status of selected drive.
AC7	Seek time out error (SKTO)	Heads did not come on track within approximately 3 seconds during a seek operation or ready to R/W was lost in "lock-on" or position mode.
AC8	Spin error (SPE)	Spindle not obtaining up-to-speed in 4Ø seconds or spindle in overspeed (sector pulses less than 593 us apart).
AC9	Write gate error	Drive sensed write gate asserted during sector pulse time, drive write locked, or drive not ready, or drive has another WDE error.
AC1Ø	Volume check (VC)	Signals that new volume (media) has been inserted into drive or power has been cycled down then up on drive.
AC11	Drive select error (DSE)	One or more drives has same number.

7.3.4 Silo Data Buffer After a Read Header

When a read header command is executed, six words are stored in the silo. The first header word contains sector address, head select, and cylinder address bit Ø (LSB) information (Figure 7-3). The second word contains the remaining bits of the cylinder address. Words 3 and 4 contain Øs and words 5 and 6 contain header CRC information. All six words can be extracted by the program using RRSI IOT.

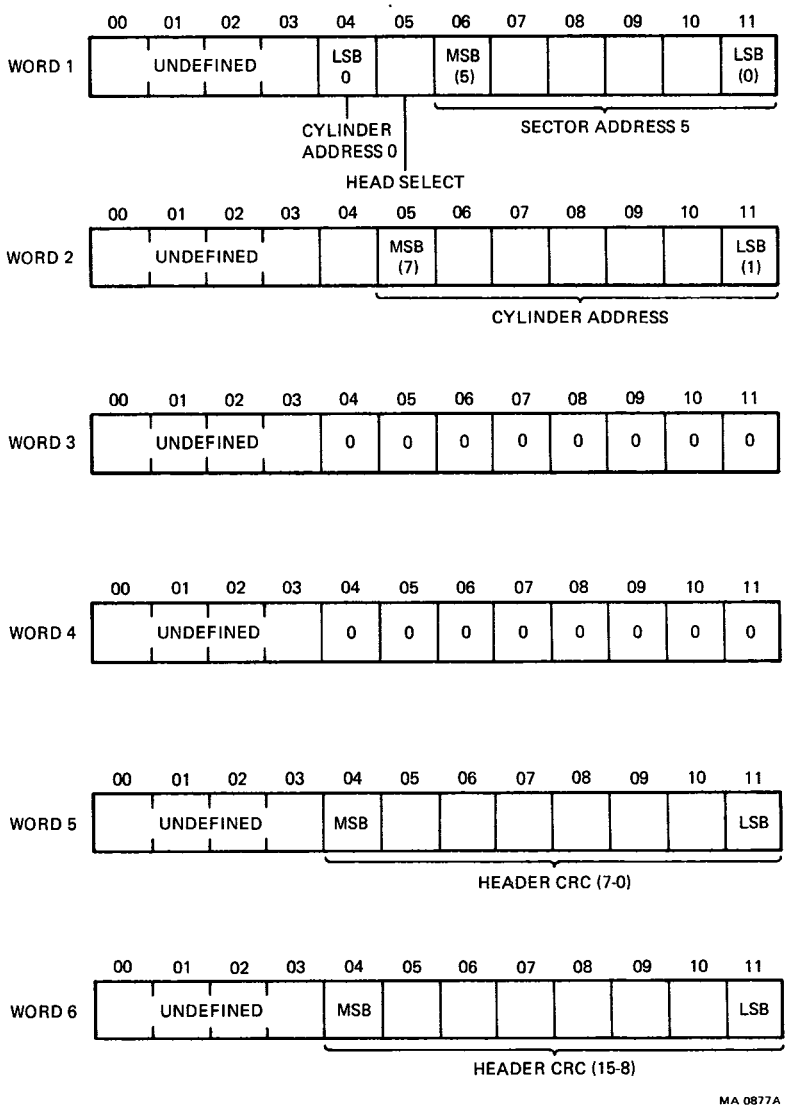


Figure 7-3 Header Words Format

7.3.5 Word Count Register

When reading or writing data, the word count is loaded with the 2's complement of the number of words to be transferred, and is then incremented by 1 as each word is transferred. Reading or writing terminates when the word count register overflows. The register can be read back into AC<Ø:11> with the RRWC instruction (IOT 6611).

The 12-bit word count register allows up to 4Ø96 data breaks (DMAs) to take place at one time. This register is loaded with the RLWC instruction (IOT 66Ø7) from AC<Ø:11> which is the 2's complement of the number of transfers specified.

7.3.6 Break MA Register

The break MA register (Figure 7-4) contains the starting memory address for word transfers between the controller and memory. BRK MA is loaded by RLMA (IOT 66Ø2) and cleared by an initialize or by loading the register with zeros. The contents of BRK MA are automatically incremented as each word is transferred between the controller and memory.

7.3.7 Sector Address Register

The sector address register (Figure 7-5) is loaded by the RLSA (IOT 66Ø5) instruction and read by RRSA (IOT 6614). It contains the sector address of the first sector to be transferred. Before executing a read or write operation, the AC<Ø:5> bits are loaded into the sector address register. The register can be read back into AC<Ø:11> with the RRSA instruction (IOT 6614).

7.3.8 Error Register

The error register (Table 7-5) is read by the RRER (IOT 661Ø) instruction. The error register can be read at anytime for error status.

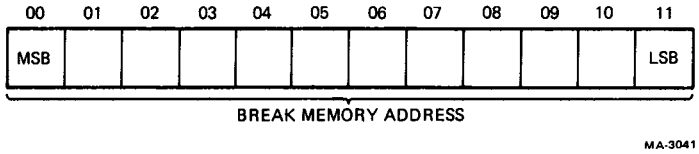


Figure 7-4 Break MA Register

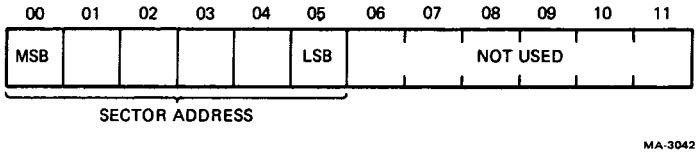
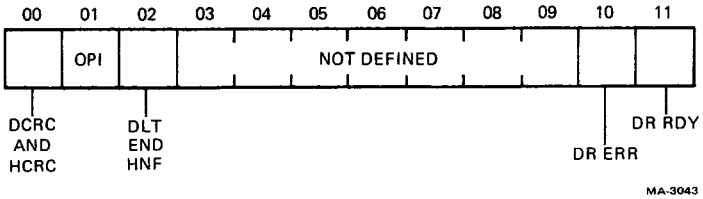


Figure 7-5 Sector Address Register

Table 7-5 Error Register Bit Definitions



Bit	Name	Function
AC0	Data CRC (DCRC) or header CRC (HCRC)	If OPI (01) is cleared and bit 00 is set, CRC error occurred in data (DCRC). If OPI (01) is set and bit 00 is also set, CRC error occurred in header (HCRC).
AC1	Operation incomplete (OPI)	When set, indicates that current command was not completed within 200 ms.
AC2	Data late (DLT) or header not found (HNF)	When OPI (bit 01) is clear and bit 02 is set, indicates that during a write operation silo was empty. Therefore, a word was not available for writing. Also indicates that during a read operation silo was full and unable to store another word from drive. When OPI (bit 01) is set and bit 02 is also set, indicates that a 200 ms timeout occurred while controller was searching for correct sector to read or write (no header compare). Bits Error 00 01 02 DLT 0 0 1 OPI 0 1 0 HNF 0 1 1 DCRC 1 0 0 HCRC 1 1 0
AC10	Drive error (DE)	This bit is tied directly to DE interface line. When set, indicates that selected drive has flagged an error. Error can be determined by executing a get status command.

Table 7-5 Error Register Bit Definitions (Cont)

Bit	Name	Function
		DE is cleared by issuing a reset command to drive.
AC11	Drive ready (DRDY)	When set, indicates that selected drive is ready to receive a command. Bit is cleared when seek operation is initiated and set when seek operation is completed.

7.4 COMMAND EXECUTION

There are eight controller commands embedded in command register B (bits 9--11). Three of these commands 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 the disk drive(s).

	Bits		
Command	9	10	11
Maintenance	Ø	Ø	Ø
Reset	Ø	Ø	1
Get status	Ø	1	Ø
Seek	Ø	1	1
Read header	1	Ø	Ø
Write data	1	Ø	1
Read data	1	1	Ø
Read data (without header check)	1	1	1

7.4.1 Maintenance Command

The maintenance command exercises the controller logic to be sure that it is operating properly (Figure 7-6). This command tests that the command registers, silo, and data paths can function correctly without the disk drive connected.

To execute the maintenance command, program the following steps:

1. Set the AC<Ø:11> to -4_{10}(7774_8). Then execute the RLWC instruction (IOT 66Ø7) to load the word count register.

2. Set AC<Ø:11> to the break address. Then execute an RLMA instruction (IOT 66Ø2) to load the break MA register.

3. Set AC<Ø:11> according to Paragraph 7.3.2 to provide maintenance mode (AC2=1) along with your choice of drive select (AC<4:5>) and the EMA field (AC<6:8>). Set AC<9:11>=Ø for the maintenance command. Then execute an RLCB instruction (IOT 66Ø4) to load command register B from AC<Ø:11> and perform the maintenance command.

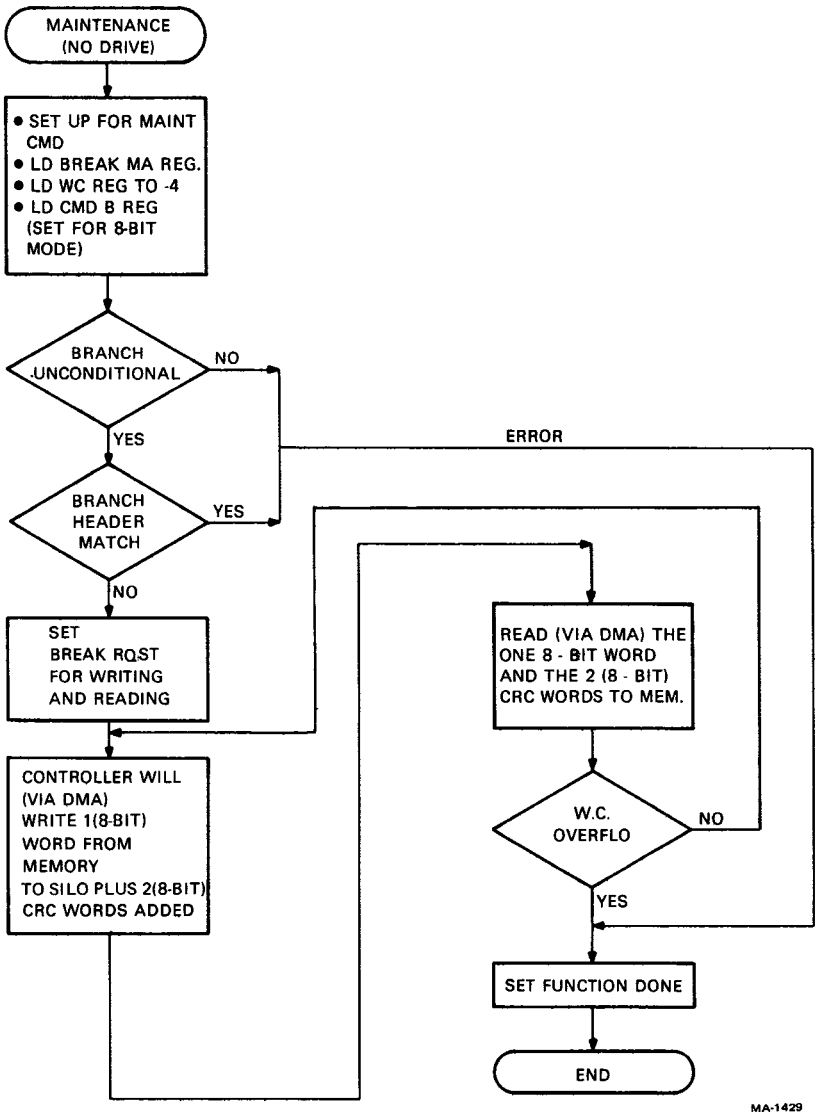


Figure 7-6 Maintenance Command, Operational Sequence

When the maintenance command is decoded the following happens:

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

2. The written data is loaded back into the silo along with two 8-bit words which are the CRC character.

3. DMAs take place causing the data (8-bit) word and the two CRC words to be written into memory.

4. This sequence is repeated until word count overflow. At this time a function done interrupt occurs. A CRC error and function done immediately raises an interrupt if a CRC error occurs.

7.4.2 Reset Command

When reset is executed, all error bits in the drive are reset. This command does not clear any registers in the controller nor does it cause the heads to move to track Ø.

NOTE: The sector address and command register A must be cleared before executing this command.

7.4.3 Get Status Command

By executing a get status command, it is possible to read the status and errors of the disk drive. The command also causes any errors present to be cleared. When this command is decoded, the drive sends the drive status word to the controller. When the function done interrupt occurs the status words can then be read from the silo into the accumulator by executing two RRSI IOTs.

NOTES: The sector address and command register A must be cleared before executing this command.

The mode bit (command register bit 2) must be set when loading the get status command into command register B.

7.4.4 Seek Command

A seek command involves two basic operations:

1. Software calculating head selection, cylinder difference, and positioning direction

2. Transmitting this information from the controller to the drive to initiate a seek.

The seek function must be specified and a difference word must be loaded into command register A to enable this function. A function done interrupt occurs when the seek command and difference word have been sent to the disk. If the difference word attempts to move the head past the innermost track 255, RLØ1 or 511, RLØ2 or the outermost track (Ø) limits, the head stops on the track limit.

NOTE: The sector address register must be cleared before executing this command. This command must be issued before a read or write unless they are already properly positioned.

7.4.5 Read Header Command

The read header command reads the first header encountered on the selected drive and stores the header information (two header words and one header CRC word) in the silo. The header words can then be extracted from the silo by reading the silo with two RRSI IOTs to determine the present head position. There are six words loaded into the silo on a read header command:

Words 1 and 2=present position
Words 3 and 4=Øs
Words 5 and 6=CRC information.

When this command is decoded the controller reads the first header encountered on the disk. The function done interrupt sets, indicating that the header words may now be read into the AC. These words (address) can be used to determine current head position or to calculate a different word to reposition the heads. The header CRC is also loaded into the controller behind the header words so that reads 5 and 6 of the silo yield these words for diagnostic purposes.

7.4.6 Write Data Command

This is a DMA command that transfers parallel data from main memory into the silo and shifts out the data in serial form to the disk drive. Prior to executing the command, it is necessary to set up the following registers:

1. Break MA register -- first memory location from which data is obtained

2. Sector address register -- desired address of first sector to be transferred

3. Word count register -- number of data words to be written, entered in 2's complement form

4. Command register A -- desired cylinder address and head select from which data is to be transferred

5. Command register B -- mode, interrupt, device, EMA function (write).

DMA transfers are continuous until word count overflows. Serial write data is continuous until the silo is emptied. At that point, the command is terminated.

Multiple sector write on one track is executed if the word count is greater than one sector. For a partial sector write, the remaining sector area is zero-filled.

Multiple sector transfers in 12-bit mode, either read or write, should not be attempted because of the odd length (8 bits) remaining at the end of a given sector. In 8-bit mode, 16 multiple sectors can be transferred.

When this command is decoded the controller begins reading successive header words and comparing them to the sector address register and command register A which contains the cylinder address. When a match is found, the header CRC is checked and if correct, that sector is written with the data starting at the address designated by the break MA register. When writing a partial sector the remainder of the sector is written with Øs. When the word count overflows, an interrupt request from function done indicates that the command is complete.

7.4.7 Read Data Command

This is a DMA command that transfers serial data from the disk drive into the silo and shifts out the data in parallel form to main memory. Before executing the command, it is necessary to set up the following registers:

1. Break MA register -- first memory location to which data is transferred

2. Sector address register -- desired address of first sector to be transferred

3. Word count register -- number of data words to be read, entered in 2's complement form

4. Command register A -- desired cylinder address from which data is to be transferred

5. Command register B -- mode, interrupt, device, EMA function (read).

DMA transfers are continuous until the word counter overflows. Serial read data is continuous until the DMA is done. At that point, the command is terminated.

When this command is decoded the controller begins reading successive header words and comparing them with the sector address register and command register A which contains the cylinder address and head select information.

When a match is found, the header CRC is checked and if correct, that sector is read. The data being read is placed into memory beginning at the address specified by the break MA register. A function done interrupt, set at word count overflow time, indicates that the transfer is complete.

If a data CRC error occurs, the CRC error flag and function done flag are both set at the completion of the transfer.

7.4.8 Read Data Without Header Check Command

When this function is decoded the controller reads the data portion of the sector following the next sector pulse. This data is read into the silo and then into memory (on second sector pulse) starting at the location specified by the break MA register. The header is neither compared nor checked for CRC errors. Data CRC is checked and the sector address (SA) is incremented. If the word count has not overflowed, the next sector is read, again without checking the header or header CRC. When word count does overflow, a function done interrupt occurs.

7.4.8.1 Track Overrun -- If word count overflow has not taken place before the end of the track has been reached, the sector address register is incremented to 50_8, an illegal sector number, causing an operation incomplete (OPI) time-out error. The sector address register then can be read showing the illegal address.

APPENDIX A
ASCII CHARACTER SETS

The ASCII character sets in this appendix contain the display codes that correspond to each of the six possible display languages. The initial display language, when the VT278 is turned on, is set at the factory. You can change the display language by issuing an SCS ESCape sequence as described in Chapter 5. Unless your VT278 system contains the foreign language support option, only the US and UK display languages are available.

Some of the characters in the foreign language character sets are actually graphic characters whose patterns are automatically retrieved from the alternate character ROM (Appendix D) for display. The display codes for the alternate character ROM also can be directly accessed if the alternate character ROM is assigned as one of the two alternate character sets defined by an SCS ESCape sequence.

ASCII octal codes ØØØ through Ø37 in each language do not produce display characters. However, octal codes ØØ1 through Ø37 can be generated from the keyboard by pressing the appropriate control character. For example, pressing and holding the CTRL key followed by pressing the S key (↑S) generates octal Ø23 (DC3).

ASCII Code		Language						
Octal	Decimal	U.S.	Dutch	French	French Canadian	German	U.K.	Meaning
000	0							NULL
001	1	↑A	↑A	↑A	↑A	↑A	↑A	SOH
002	2	↑B	↑B	↑B	↑B	↑B	↑B	STX
003	3	↑C	↑C	↑C	↑C	↑C	↑C	ETX
004	4	↑D	↑D	↑D	↑D	↑D	↑D	EOT
005	5	↑E	↑E	↑E	↑E	↑E	↑E	ENQ
006	6	↑F	↑F	↑F	↑F	↑F	↑F	ACK
007	7	↑G	↑G	↑G	↑G	↑G	↑G	BEL
010	8	↑H	↑H	↑H	↑H	↑H	↑H	BS
011	9	↑I	↑I	↑I	↑I	↑I	↑I	HT
012	10	↑J	↑J	↑J	↑J	↑J	↑J	LF
013	11	↑K	↑K	↑K	↑K	↑K	↑K	VT
014	12	↑L	↑L	↑L	↑L	↑L	↑L	FF
015	13	↑M	↑M	↑M	↑M	↑M	↑M	CR
016	14	↑N	↑N	↑N	↑N	↑N	↑N	SO
017	15	↑O	↑O	↑O	↑O	↑O	↑O	SI
020	16	↑P	↑P	↑P	↑P	↑P	↑P	DLE
021	17	↑Q	↑Q	↑Q	↑Q	↑Q	↑Q	DC1
022	18	↑R	↑R	↑R	↑R	↑R	↑R	DC2
023	19	↑S	↑S	↑S	↑S	↑S	↑S	DC3
024	20	↑T	↑T	↑T	↑T	↑T	↑T	DC4
025	21	↑U	↑U	↑U	↑U	↑U	↑U	NAK
026	22	↑V	↑V	↑V	↑V	↑V	↑V	SYN
027	23	↑W	↑W	↑W	↑W	↑W	↑W	ETB
030	24	↑X	↑X	↑X	↑X	↑X	↑X	CAN
031	25	↑Y	↑Y	↑Y	↑Y	↑Y	↑Y	EM
032	26	↑Z	↑Z	↑Z	↑Z	↑Z	↑Z	SUB
033	27	↑[	↑ij	↑é	↑é	↑ä	↑[	ESC
034	28	↑\	↑<	↑ù		↑ö	↑\	FS
035	29	↑]		↑¨	↑=	↑ü	↑]	GS
036	30	↑^	↑´	↑^		↑β	↑^	RS
037	31	↑/		↑:			↑/	US

MA-9084

ASCII Code		Language					
Octal	Decimal	U.S.	Dutch	French	French Canadian	German	U.K.
040	32	space	space	space	space	space	space
041	33	!	!	!	!	!	!
042	34	"	"	"	"	"	"
043	35	#	£	£	£	#	£
044	36	$	$	$	$	$	$
045	37	%	%	%	%	%	%
046	38	&	&	&	&	&	&
047	39	'	'	'	'	'	'
050	40	(	(	(	(	(	(
051	41	)	)	)	)	)	)
052	42	*	*	*	*	*	*
053	43	+	+	+	+	+	+
054	44	'	'	'	'	'	'
055	45	-	-	-	-	-	-
056	46	.	.	.	.	.	.
057	47	/	/	/	/	/	/
060	48	0	0	0	0	0	0
061	49	1	1	1	1	1	1
062	50	2	2	2	2	2	2
063	51	3	3	3	3	3	3
064	52	4	4	4	4	4	4
065	53	5	5	5	5	5	5
066	54	6	6	6	6	6	6
067	55	7	7	7	7	7	7
070	56	8	8	8	8	8	8
071	57	9	9	9	9	9	9
072	58	:	:	:	:	:	:
073	59	;	;	;	;	;	;
074	60	<	<	<	<	<	<
075	61	=	=	=	=	=	=
076	62	>	>	>	>	>	>
077	63	?	?	?	?	?	?

MA-9085

ASCII Code		Language					
Octal	Decimal	U.S.	Dutch	French	French Canadian	German	U.K.
100	64	@	3/4	à	à	§	@
101	65	A	A	A	A	A	A
102	66	B	B	B	B	B	B
103	67	C	C	C	C	C	C
104	68	D	D	D	D	D	D
105	69	E	E	E	E	E	E
106	70	F	F	F	F	F	F
107	71	G	G	G	G	G	G
110	72	H	H	H	H	H	H
111	73	I	I	I	I	I	I
112	74	J	J	J	J	J	J
113	75	K	K	K	K	K	K
114	76	L	L	L	L	L	L
115	77	M	M	M	M	M	M
116	78	N	N	N	N	N	N
117	79	O	O	O	O	O	O
120	80	P	P	P	P	P	P
121	81	Q	Q	Q	Q	Q	Q
122	82	R	R	R	R	R	R
123	83	S	S	S	S	S	S
124	84	T	T	T	T	T	T
125	85	U	U	U	U	U	U
126	86	V	V	V	V	V	V
127	87	W	W	W	W	W	W
130	88	X	X	X	X	X	X
131	89	Y	Y	Y	Y	Y	Y
132	90	Z	Z	Z	Z	Z	Z
133	91	[	ij	o	o	Ä	[
134	92	\	1/2	ç	ç	Ö	\
135	93	]	|	§	§	Ü	]
136	94	^	^	^	^	^	^
137	95	_	_	_	_	_	_

MA-9086

ASCII Code		Language					
Octal	Decimal	U.S.	Dutch	French	French Canadian	German	U.K.
140	96	`	`	`	`	`	`
141	97	a	a	a	a	a	a
142	98	b	b	b	b	b	b
143	99	c	c	c	c	c	c
144	100	d	d	d	d	d	d
145	101	e	e	e	e	e	e
146	102	f	f	f	f	f	f
147	103	g	g	g	g	g	g
150	104	h	h	h	h	h	h
151	105	i	i	i	i	i	i
152	106	j	j	j	j	j	j
153	107	k	k	k	k	k	k
154	108	l	l	l	l	l	l
155	109	m	m	m	m	m	m
156	110	n	n	n	n	n	n
157	111	o	o	o	o	o	o
160	112	p	p	p	p	p	p
161	113	q	q	q	q	q	q
162	114	r	r	r	r	r	r
163	115	s	s	s	s	s	s
164	116	t	t	t	t	t	t
165	117	u	u	u	u	u	u
166	118	v	v	v	v	v	v
167	119	w	w	w	w	w	w
170	120	x	x	x	x	x	x
171	121	y	y	y	y	y	y
172	122	z	z	z	z	z	z
173	123	{	¨	é	é	ä	{
174	124	|	f	ù	ù	ö	|
175	125	}	1/4	è	è	ü	}
176	126	~	´	¨	¨	ß	~
177	127	DEL	DEL	DEL	DEL	DEL	DEL

MA-9087

APPENDIX B
ENGLISH AND FOREIGN LANGUAGE KEYBOARDS

The VT278 keyboards for the English, Dutch, French, French-Canadian and German languages are shown in Figures B-1 through B-5, respectively.

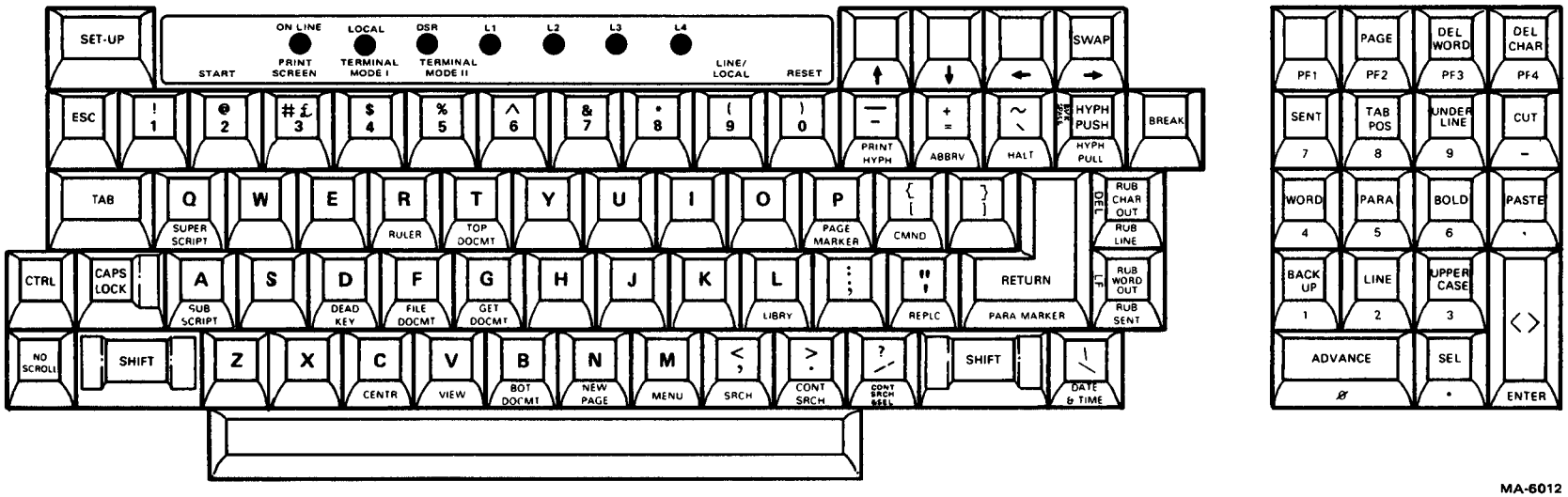


Figure B-1 English Keyboard

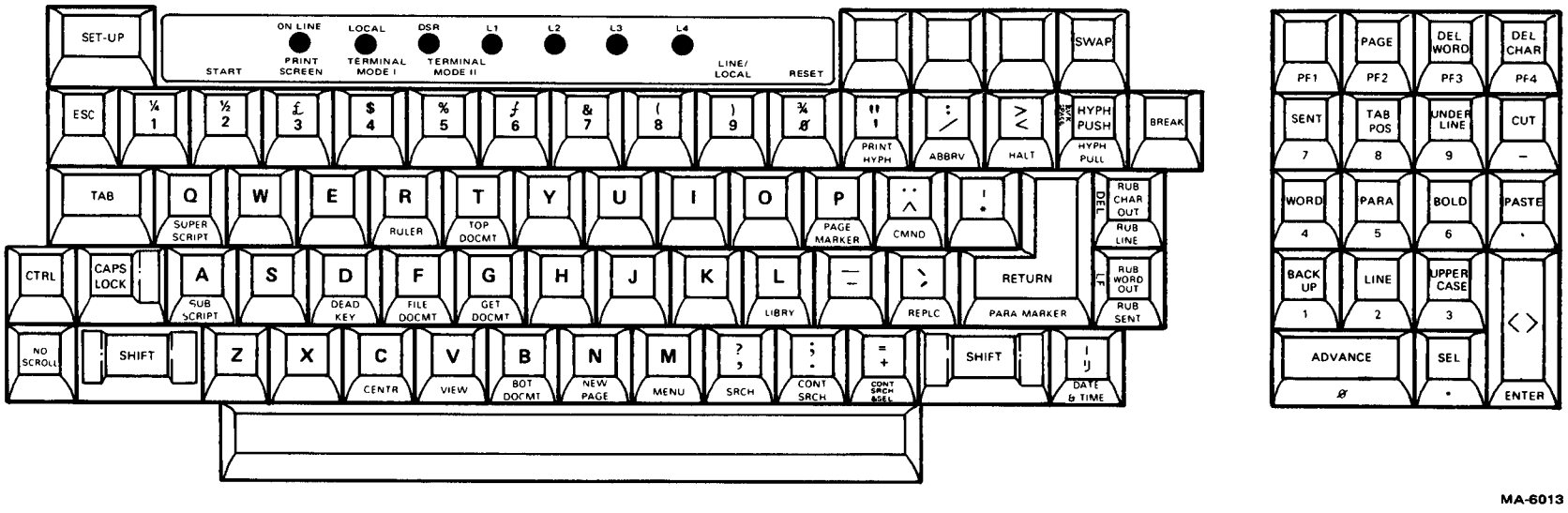
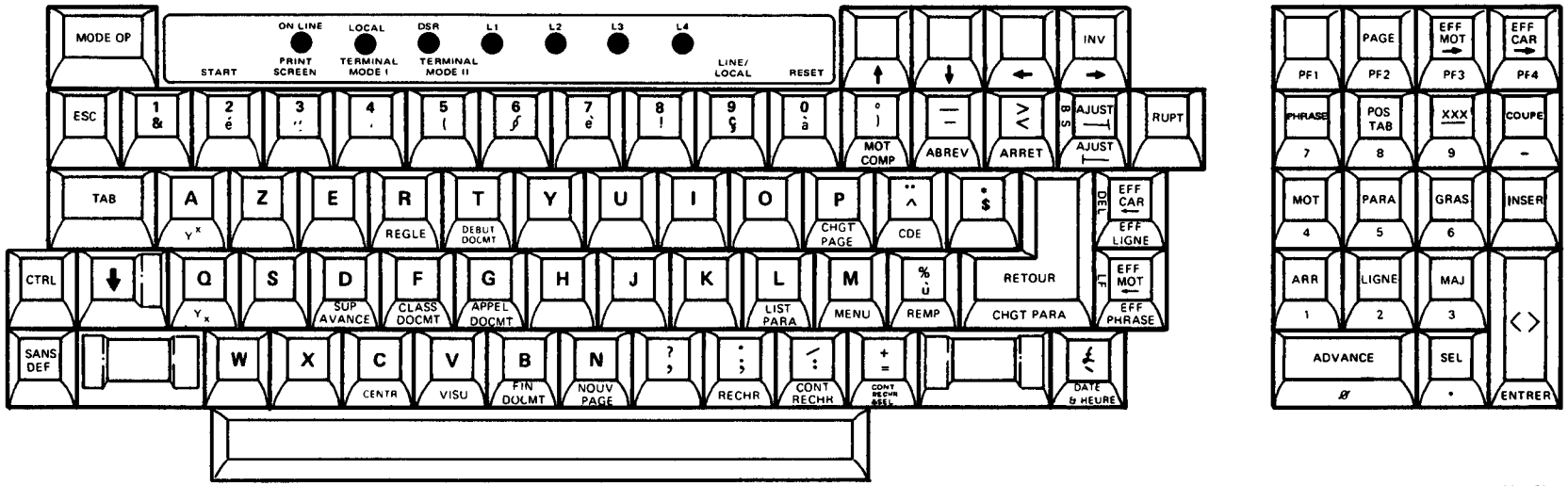


Figure B-2 Dutch Keyboard



MA-6015

Figure B-3 French Keyboard

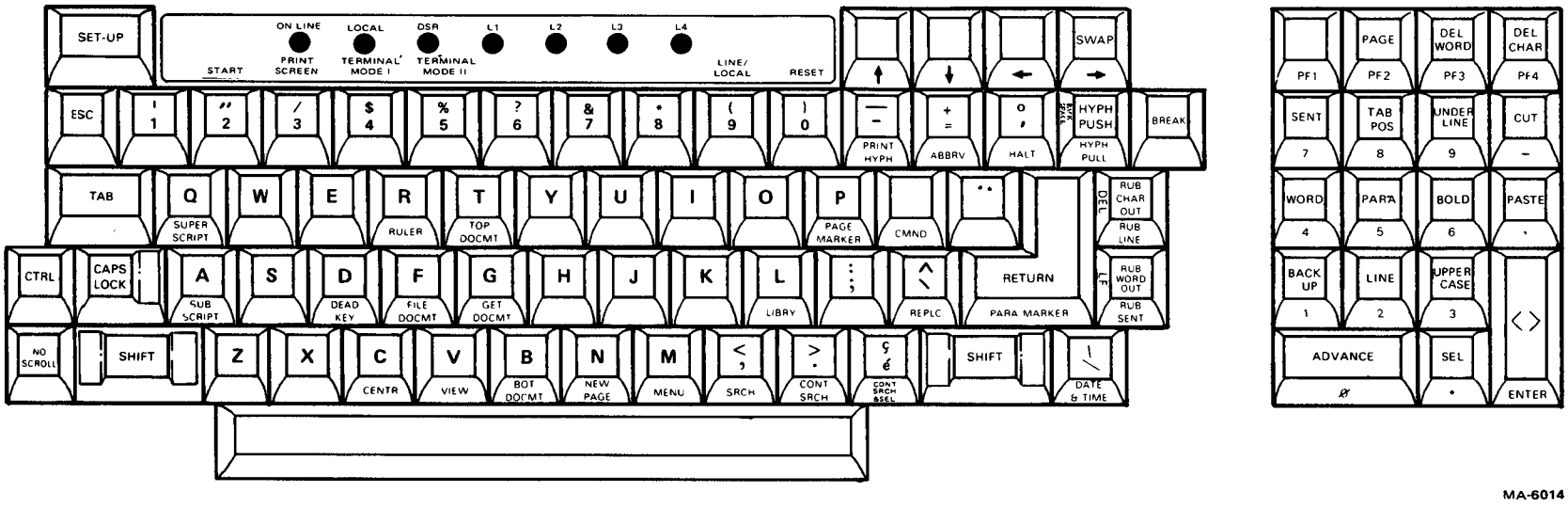


Figure B-4 French-Canadian Keyboard

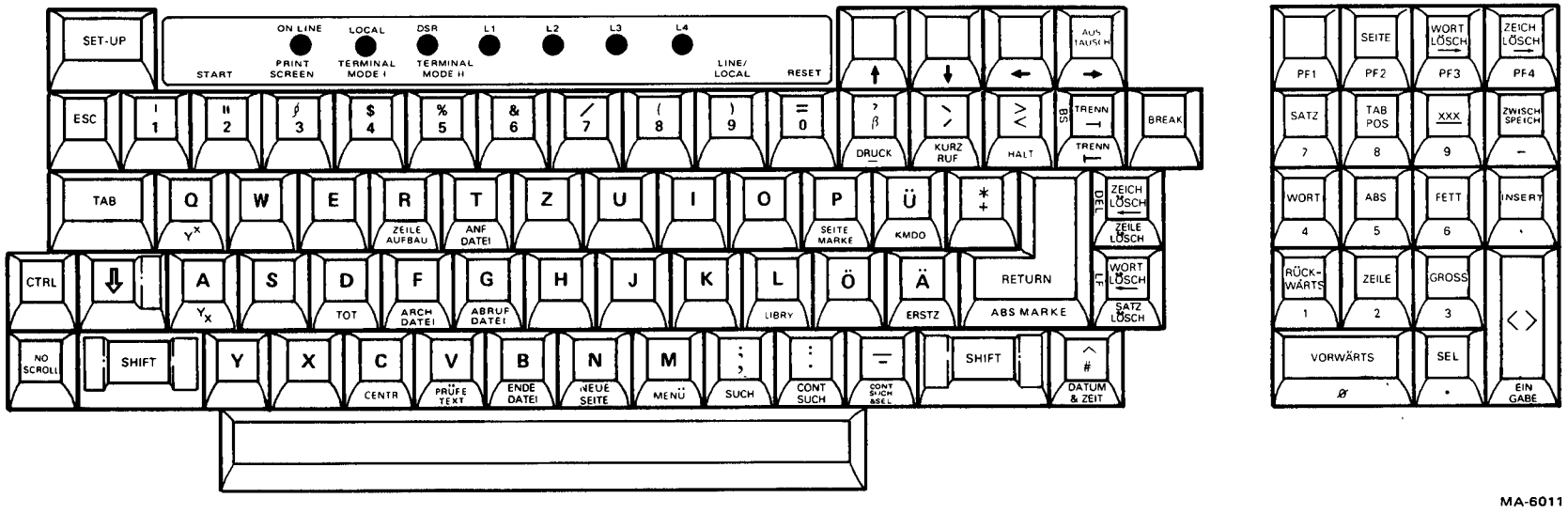


Figure B-5 German Keyboard

APPENDIX C
SPECIAL GRAPHICS

You can assign the special graphics character set as one of the two alternate character sets by executing an SCS (select character set) ESCape sequence as defined in Chapter 5. The special graphics characters are shown in Figure C-1.

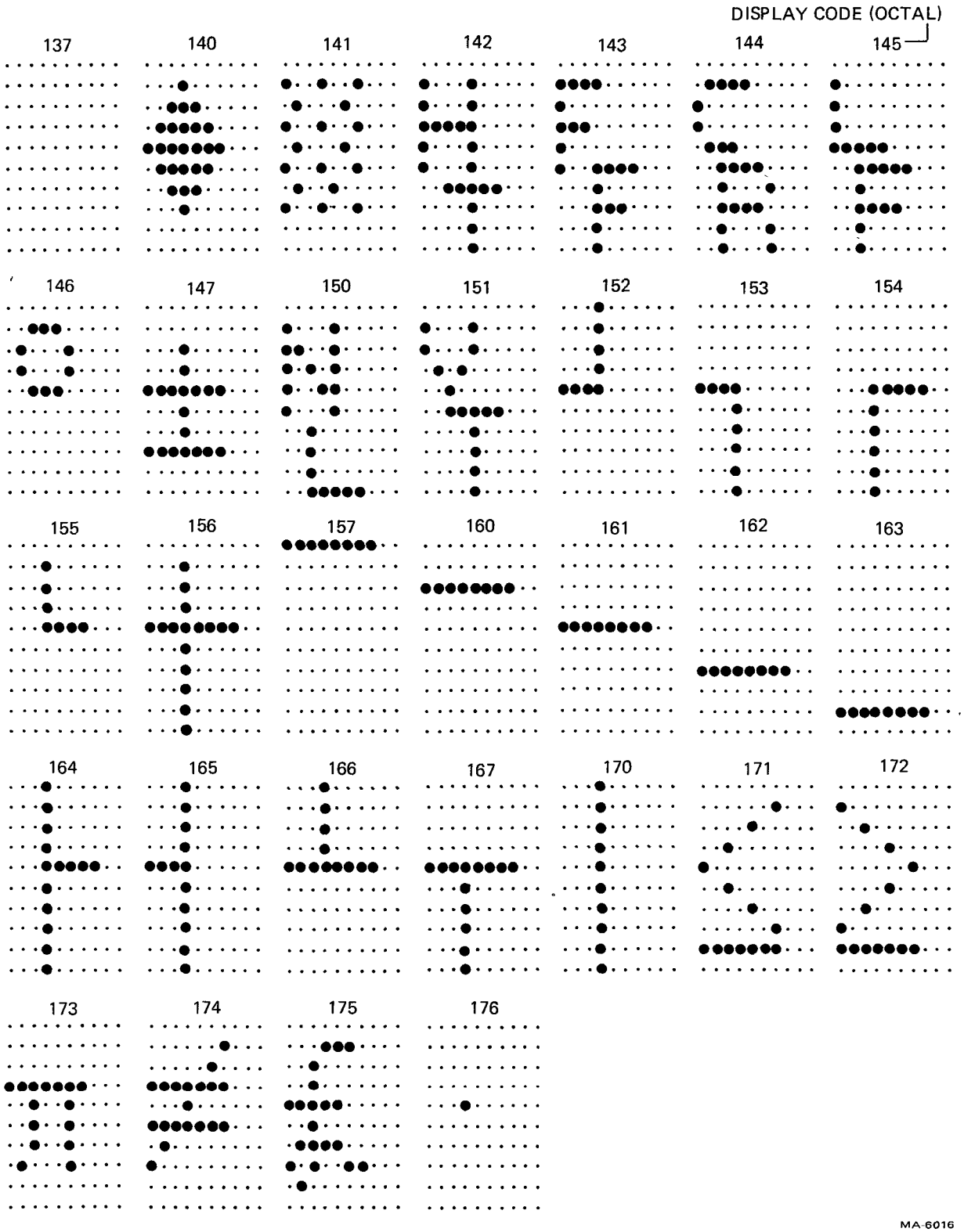


Figure C-1 Special Graphics

APPENDIX D
ALTERNATE CHARACTER ROM, STANDARD GRAPHICS

You can assign this character set as one of the two available character sets by executing an SCS (select character set) ESCape sequence as defined in Chapter 5. The standard graphics characters in the alternate character ROM are shown in Figure D-1.

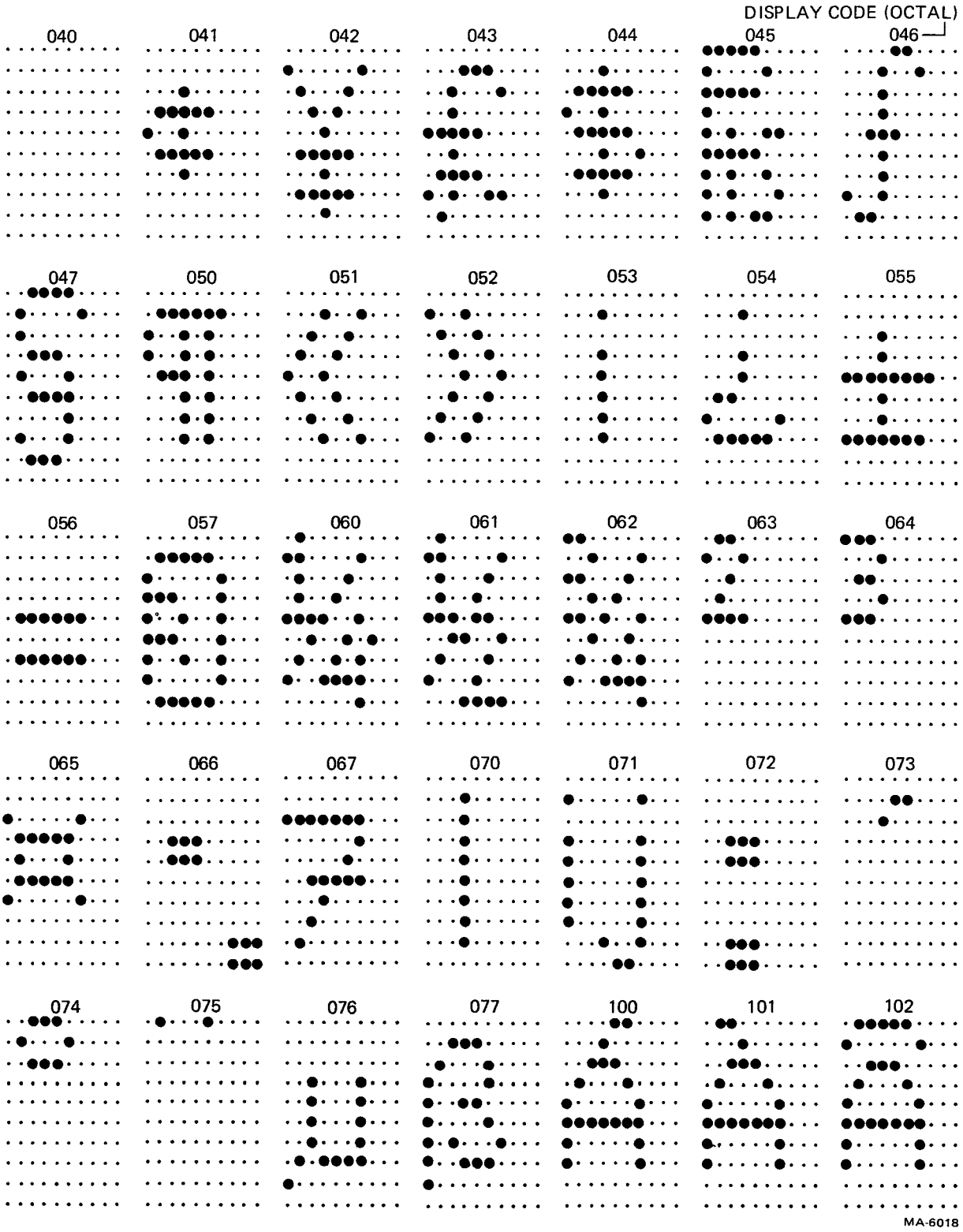


Figure D-1 Alternate Character ROM, Standard Graphics (Sheet 1 of 3)

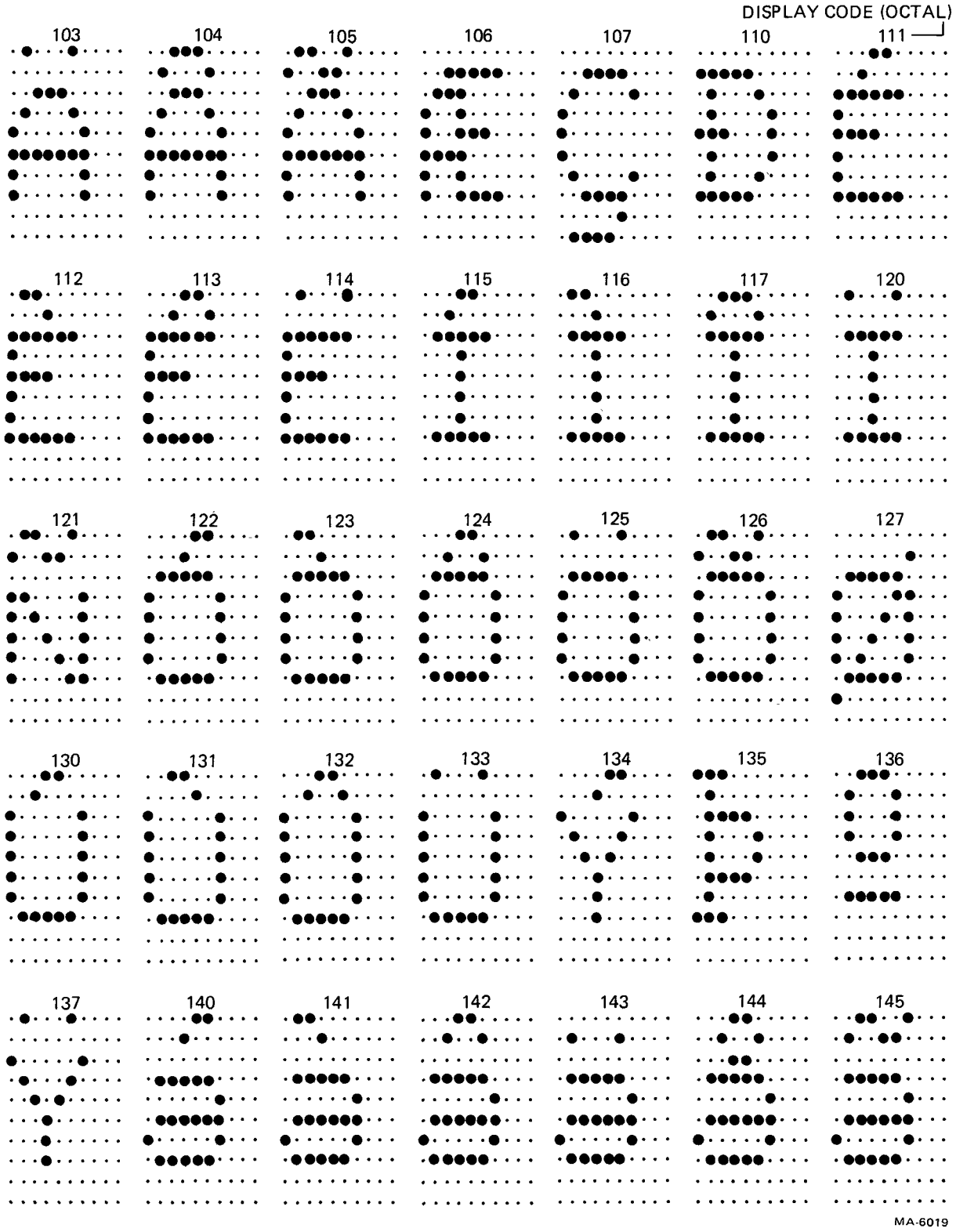


Figure D-1 Alternate Character ROM, Standard Graphics (Sheet 2 of 3)

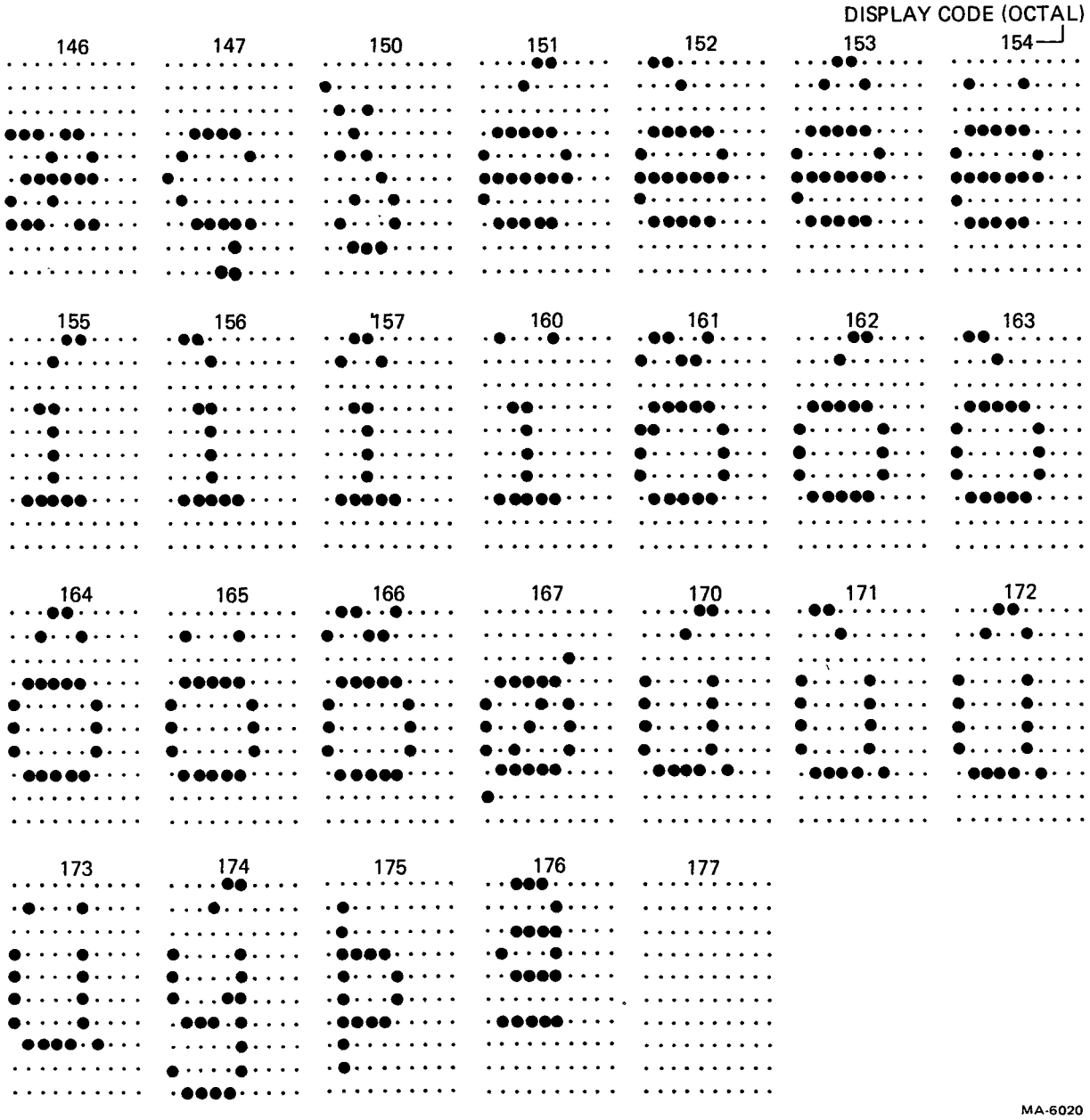


Figure D-1 Alternate Character ROM, Standard Graphics (Sheet 3 of 3)

APPENDIX E
ALTERNATE CHARACTER ROM, SPECIAL GRAPHICS

You can assign this character set as one of the two available character sets by executing an SCS (select character set) ESCape sequence as defined in Chapter 5. The special graphics characters in the alternate character ROM are shown in Figure E-1.

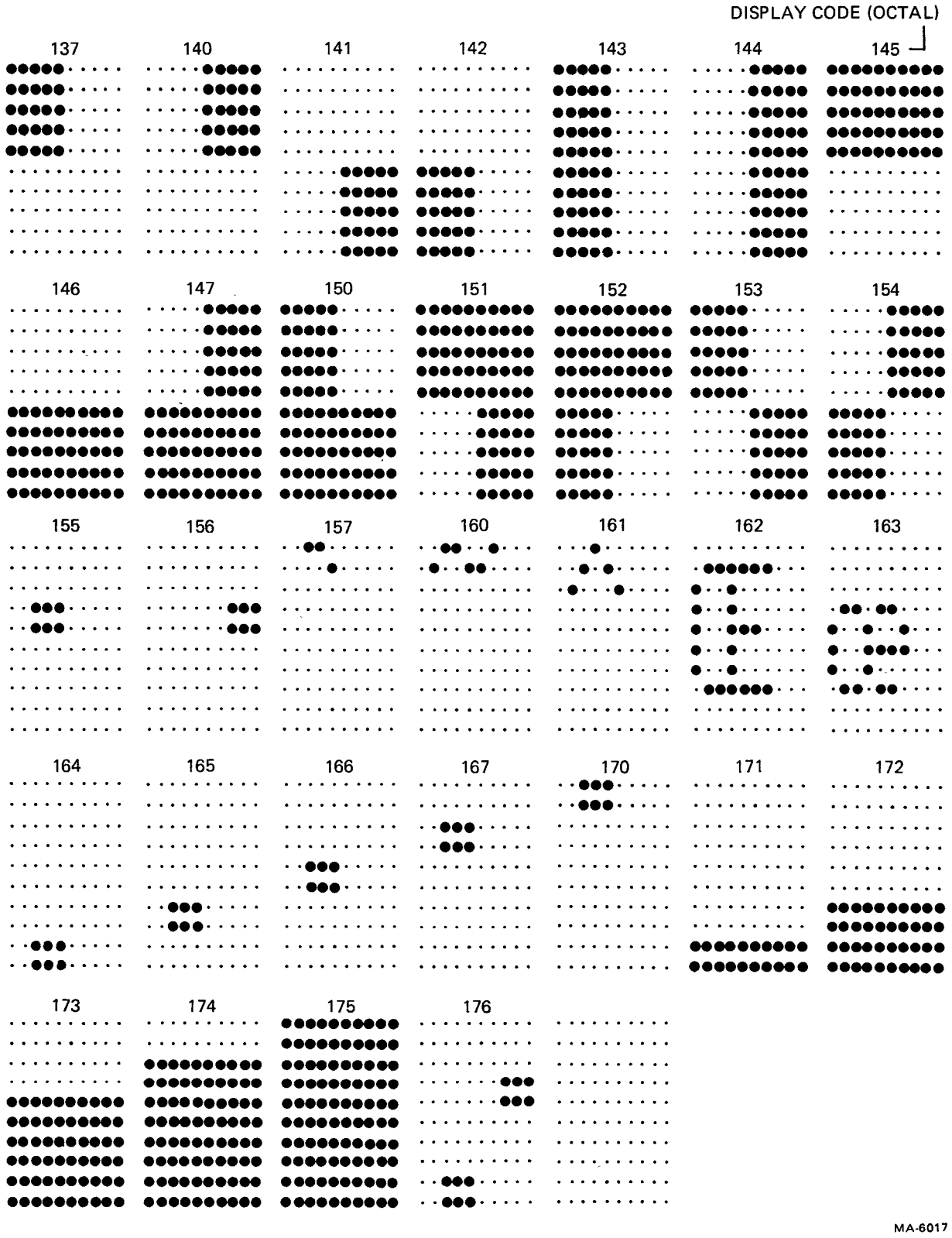


Figure E-1 Alternate Character ROM, Special Graphics

APPENDIX F
DETAILED SPECIFICATIONS

F.1 VT278 SPECIFICATIONS

Dimensions

Monitor	
Height	36.83 cm (14.5 in.)
Width	45.72 cm (18 in.)
Depth	36.2Ø cm (14.25 in.)
Keyboard	
Height	8.89 cm (3.5 in.)
Width	45.72 cm (18 in.)
Depth	2Ø.32 cm (8 in.)
Minimum table depth	51.4 cm (2Ø.25 in.)

Weight

Monitor	13.6 Kg (3Ø lbs)
Keyboard	2.Ø kg (4.5 lbs)
Shipping weight	18.6 Kg (41 lbs)

Environmental

Operating	
Temperature	1Ø° to 4Ø° C (5Ø° to 1Ø4° F)
Relative humidity	1Ø% to 9Ø%
Maximum wet bulb	28° C (82° F)
Minimum dew point	2° C (36° F)
Altitude	2.4 km (8ØØØ ft)
Nonoperating	
Temperature	-4Ø° to 66° C (-4Ø° to 151° F)
Relative humidity	Ø% to 95%
Altitude	9.1 km (3Ø,ØØØ ft)

Power

Line voltage	9Ø--128 V rms single phase, 3 wire 18Ø--256 V rms single phase, 3 wire (switch-selectable)

Line frequency	47--63 Hz
Current	2.2 A rms max @ 115 V rms 1.1 A rms max @ 230 V rms
Input power	250 VA apparent 150 W max
Current limiting	3 A normal blow fuse
Power cord	Detachable, 3-prong, 1.9 m (6 ft)
Display	
CRT	12-inch diagonal measure, P4 phosphor
Format (selectable)	24 lines X 80 characters or 14 lines X 132 characters
Character	7 X 9 dot matrix with descenders
Character size	80 column mode 3.35 mm X 2.0 mm (0.132 inch X 0.078 inch) 132 column mode 3.35 mm X 1.3 mm (0.132 inch X 0.051 inch)
Active display size	202 mm X 115 mm (8 inch X 4.5 inch)
Character set	96-character ASCII subset standard 128 additional as option
Cursor type	Blinking solid block
Keyboard	
General	83-key detachable unit with 1.9 m (6 ft) coiled cord
Keyboard layout	65-keys similar to standard typewriter keyboard with 18-key numeric keypad
Numeric keypad	18-keys with period, comma, minus, enter, and four general-purpose function keys
Visual indicators	Seven LEDs: on line, local, and data set ready, and four user-programmable.

Audible signals	Keyclick: sound simulates typewriter. Bell sounds upon receipt of BEL code
I/O Interconnect	
Printer port	25-pin male connector (Figure F-1)
RX278 floppy disk interface	37-pin female connector (Figure F-2)
DP278 communications option	Two 25-pin male connectors, (Figure F-3)
RL278 hard disk interface	37-pin male connector (Figure F-4)

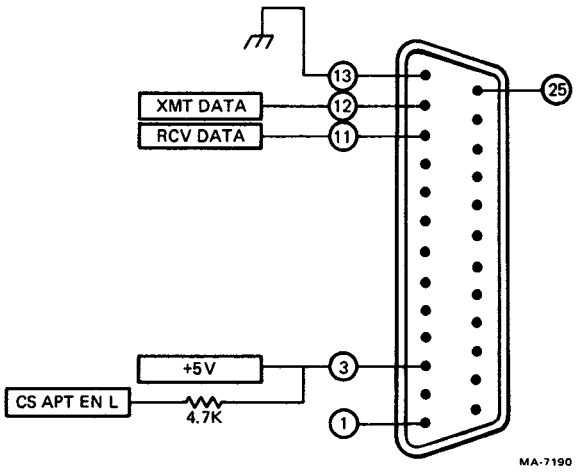


Figure F-1 VT278 Printer Port Connector

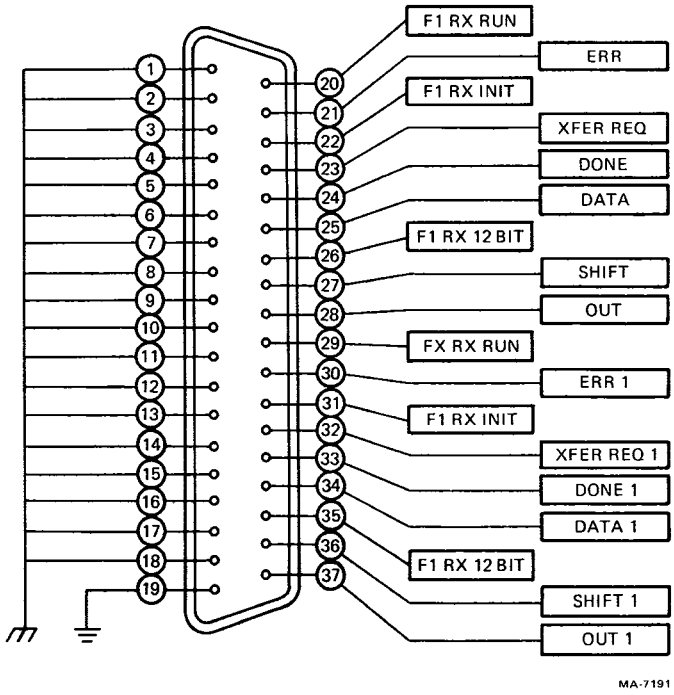


Figure F-2 RX278 Floppy Drive Interface Connector

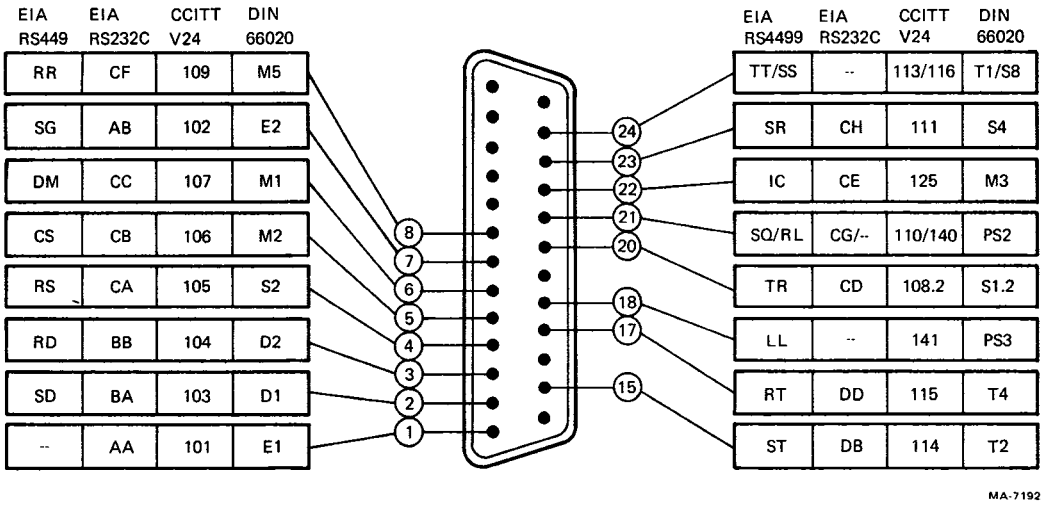


Figure F-3 DP278 Communications Options Connector (Port0/Port 1)

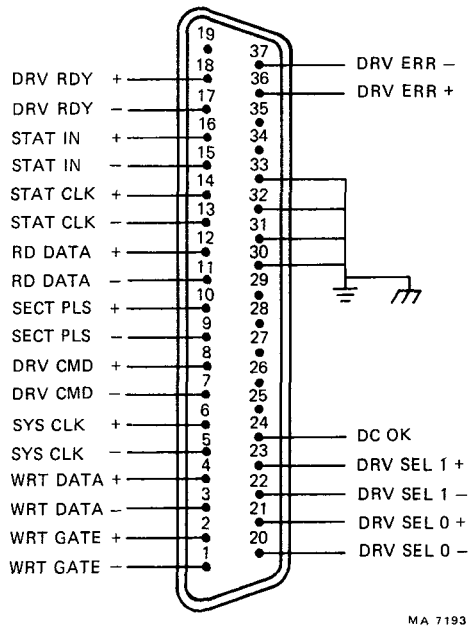


Figure F-4 RL278 Disk Cartridge Interface Connector

F.2 RXØ2 DISKETTE DRIVE SPECIFICATIONS

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

NOTE: The above error rates only apply to DEC approved media that are properly cared for. Seek error and soft data errors are usually attributable to random effects in the head/media interface, such as electrical noise, dirt, or dust. Both are called "soft" errors if the error is recoverable in 1Ø additional tries or less. "Hard" errors cannot be recovered. Seek error retries should be preceded by a recalibrate.

Drive performance

Capacity	Recording	8-bit bytes	12-bit words
Per diskette	FM (single density)	256,256	128,128
	MFM (double density)	512,512	256,256
Per track	FM	3328	1664
	MFM	6656	3328
Per sector	FM	128	64
	MFM	256	128

Data transfer rate	
Diskette to controller buffer	4 us/data bit (FM); 2 us/data bit (MFM)
Buffer to CPU interface	1.2 us/bit

NOTE: RX278 interface can operate in 8- or 12-bit modes under software control.

Track-to-track move	6 ms/track maximum
Head settle time	25 ms maximum
Rotational speed	360 rpm ±2.5%; 166 ms/rev nominal
Recording surfaces/disk	1
Tracks/disk	77 (0--76) or ($0--114_8$)
Sectors/track	26 (1--26) or ($0--32_8$)
Recording technique	Double frequency (FM) or modified MFM
Bit density maximum on inner track	3200 bpi (FM) or modified (MFM)
Track density	48 tracks/inch
Average access	262 ms, computed as follows:

Seek	Settle	Rotate

77 tks/3 X 6 ms + 25 ms + 166 ms/2 = 262 ms

ENVIRONMENTAL CHARACTERISTICS

Temperature	
RX02, operating	15° to 32° C (59° to 90° F) ambient; max. temp. gradient = 11° C/hr (20° F/hr)
RX02, nonoperating	-35° to +60° C (-30° to +140° F)
Media, nonoperating	-35° to +52° C (-30° to +125° F)

NOTE: Media temperature must be within operating temperature range before use.

Heat dissipation (RX02 system)	less than 225 btu/hr
Relative humidity	
RX02, operating	25° C (77° F) maximum wet bulb 2° C (36° F) minimum dew point 20% to 80% relative humidity
RX02, nonoperating	5% to 98% relative humidity (no condensation)
Media, nonoperating	10% to 80% relative humidity
Magnetic field	Media exposed to a magnetic field strength of 50 oersteds or greater may lose data.

Interface modules	
Operating temperature	5° to 50° C (41° to 122° F)
Relative humidity	10% to 90%
Maximum wet bulb	32° C (90° F)
Minimum dew point	2° C (36° F)
Electrical	
RX02 power consumption	5 A @ +5 Vdc, 25 W; 0.14 A @ -5 Vdc, 0.7 W; 1.3 A @ +24 Vdc, 31 W
AC power	4 A @ 115 Vac 2 A @ 230 Vac

F.3 RL02 DISK DRIVE SPECIFICATIONS

Physical and Environmental

Width	Compatible with 48.26 cm (19 in) RETMA rack
Depth	63.5 cm (25 in) behind bezel
Height	26.52 cm (10.44 in)
Weight	34.02 kg (75 lb)
Mounting	Drive mounts on chassis slides
Power source	90--127 Vac (47--63 Hz) 180--254 Vac (47--63 Hz) (manually-selectable)
Input power	160 W max @ 115 Vac, 60 Hz
Power factor	Greater than 0.85
Starting current	3.5 A rms max @ 90 Vac/47--63 Hz 5.0 A rms max @ 127 Vac/47--63 Hz 1.75 A rms max @ 180 Vac/47--63 Hz 2.5 A rms max @ 254 Vac/47--63 Hz
Heat dissipation	600 Btu/hr max
Power cord and connector	Molded line cord compatible with drive operating voltage, and 861 power control for 120 Vac attaches to drive. Power cord is 2.74 m (9 ft) long; plug is NEMA 5-15P. 230 Vac plug to attach to high voltage drives is NEMA 6-15P
Safety	RL02 disk drive is UL listed and CSA certified.
Interlocks	Used where potential exists for damage to drive, media, operators, or service personnel.

Temperature/Humidity	
Operating	
Temperature	10° C (50° F) to 40° C (104° F)
Relative humidity	10% to 90%
Maximum wet bulb	28° C (82° F)
Minimum dew point	2° C (36° F)
Nonoperating	
Temperature	-40° C (-40° F) to 66° C (151° F)
Relative humidity	10% to 95% noncondensing
Altitude	
Operating	2440 m (8,000 ft) max
Nonoperating	9.1 km (30,000 ft) max
Shock	
Operating	Half sine shock pulse of gravity peak and 10 ±3 ms duration applied once in either direction of 3 orthagonal axes (3 pulses total).
Nonoperating	Half sine shock pulses of 40 gravity peak and 30 ±10 ms duration perpendicular to each of six package surfaces.
Vibration	
Operating	
Sinusoidal vibration	(sweep rate 1 octave/min) 5--50 Hz, 0.002 in displacement amplitude 50--500 Hz, 0.25 gravity peak 500--50 Hz, 0.25 gravity peak 50--5 Hz, 0.002 in displacement amplitude
Nonoperating	
Vertical axis excitation	1.40 gravity rms overall from 10--300 Hz, Power spectral density of 0.029 g^2/Hz from 10 to 50 Hz, with 8 dB/octave rolloff from 50--300 Hz.
Longitudinal and lateral axis excitation	0.68 gravity rms overall from 10--200 Hz, Power spectral density of 0.007 g^2/Hz from 10--50 Hz, with 8 dB/octave rolloff from 50--200 Hz.
EMI	Meets DEC Standard 102, Section 7.

Dust	The drive will operate in an ambient atmosphere of less than 5 million particles Ø.5 microns or larger per cubic foot of air. The drive is intended to run in a light industry or cleaner environment.
Attitude	
Maximum pitch	±15 degrees
Maximum roll	±15 degrees

Operational Specifications

General	
Linear bit density	3725 bits/in at track 255 (decimal)
16 bit words per sector	128
Number of sectors	4Ø per surface
Track density	125/in
Number of tracks	256 per surface
Number of surfaces	2
Formatted capacity	5.2 (megabytes)
Encoding method	MFM
Transfer rate (unbuffered values)	
Bit rate	4.1 megabits/second ±1%
Bit cell width	244 ns ±1%
Words (16 bit)	256 kilowords/second ±1%
Latency	
Rotational frequency	24ØØ rpm
Average latency	12.5 ms ±Ø.25 percent
Maximum latency	25.Ø ms ±Ø.25 percent
Seek Time	
Average	55 ms max (17Ø tracks)
One track	15 ms max
Maximum	1ØØ ms max (512 tracks)
Head switching time	8 ms max
Start/Stop Time	
Start	45 seconds
Stop	3Ø seconds

Disk Cartridge Specifications

Operating environment	Operating temperature of air surrounding cartridge is within 4Ø° to 48° C (4Ø° to 12Ø° F) range at relative humidity of 8 to 8Ø percent. Wet bulb reading shall not exceed 25° C (78° F). Before a cartridge is used, it should be kept in its cover for a minimum of

	two hours in the same environment as the disk drive in operation. Above specified range does not necessarily apply to disk drive.
Storage environment	Storage temperature is from -40° to 65° C (-40° to 150° F). Wet bulb reading not exceeding 29° C (85° F). For wet bulb temperatures between 0.56° and 29° C (33° and 85° F) disk cartridge shall be able to withstand a relative humidity of 8--80%. Stray magnetic field intensity shall not exceed 50 oersteds.
Dimensions (cartridge)	External diameter of top cover is 38.35 cm (15.1 in).
	External diameter of protection cover is 37.03 cm (14.58 in).
	External height of cartridge resting on bottom surface is 6.19 cm (2.44 in).
Maximum speed	Rotating parts of disk cartridge can withstand effect of stress at speed of 2500 rev/min.
Track geometry	512 discrete concentric tracks per data surface.
Data track identification	Three-digit decimal number (000--511) that numbers data tracks consecutively starting at outermost data track of each data surface.
Data surface identification	Numbered 0 and 1 starting with uppermost surface and corresponding with head numbers.
Cylinder address	A cylinder is defined as the data tracks on the data surface, one track per surface, with a common data track identification.
Data track address	A 16-bit word, where bits 0--5 define a binary sector number, bit 6 defines surface, and bits 7--15 define a binary cylinder address. This information is in word 1 of each sector's header.

APPENDIX G
PROGRAMMING CONSIDERATIONS

G.1 VT278 PROGRAMMING DIFFERENCES

The VT278 is upward compatible with earlier PDP-8 type computers. The following paragraphs highlight the significant differences between the VT278 and its predecessors.

G.1.1 Normal Interrupt Operation

The "Skip on Flag" IOT (6XY1) instructions in the VT278 clear a flag when the flag is found to be set. IOT skip instructions on earlier PDP-8 type computers did not clear the flag. Most programs do not find this difference a problem. However, if a program relies on the flag to still be set after the skip, the program needs to be changed.

The following example shows code that fails on a VT278 but runs on the earlier machines.

```
TST1,   6031        /Test keyboard flag
        JMP OTHER   /Not set, try something else
        . . . .     /Come here when skip occurs
        . . . .
TST2,   6031        /Now try flag again
        HLT         /This time it didn't skip as the
                    /first 6031 cleared it.
```

G.1.2 Interrupt Enables

Previously, devices 03, 04 and 32, 33 had a single interrupt enable for each device pair. The VT278 has a separate interrupt enable for each device of the pair and separate instructions for setting and clearing each.

Instruction	VT78	VT278
6035	Load interrupt enable for both 03 and 04 from AC11.	Load interrupt enable for 03 only from AC11.
6045	Skip if either device 03 or 04 is causing an interrupt.	Load interrupt enable for 04 only from AC11.

6325	Load interrupt enable for both 32 and 33 from AC11.	Load interrupt enable for 32 only from AC11.
6335	Skip if either device 32 or 33 is causing an interrupt.	Load interrupt enable for 33 only from AC11.

Interrupt service routines that use the 6045(6335) skip need to be altered. In general individual interrupt enables should cause no problems except when programs attempt to disable interrupts from device 04(33) by issuing IOT 6035.

G.1.3 IOT 6030 and 6320

These instructions previously clear the TTI/TTI2 flags. On the VT278, the instructions set the flags.

G.1.4 Stack Operations

VT278 stack instructions should not produce any incompatibility problems unless the operating software arbitrarily enables a (nonexistent) KT8A.

G.1.5 R3L Instruction

This new instruction rotates the AC three positions to the left, leaving the link unchanged by directly moving AC0 into AC11. This was previously an illegal instruction. It should pose no problems unless it was used to identify the type CPU on which a program was running. It may now indicate the 'wrong' type CPU.

G.1.6 WSR and OSR Instructions

The WSR instruction was previously illegal. On the VT278, it can be used to clear the AC. The OSR instruction is used on the VT278 to read status information. There is no switch register on the VT278.

G.1.7 The HLT Instruction

The HLT instruction or any group 2 microinstruction that includes a HLT should never be executed in a user program. The effect is a trap into control panel memory, entry into SET-UP mode, and the display of a message of the form:

```
HLT XXXX YYYY ; AC,MQ,SP1,SP2 AAAA MMMM SSSS PPPP
```

Where:

XXXX = flag word at time HLT was executed

YYYY = program counter +1 (instruction address following HLT)

AAAA = value in AC

MMMM = value in MQ

SSSS = value in SP1

PPPP = value in SP2

The only recovery from this state is to reboot.