reference manual

HEWLETT PACKARD

intelligent graphics 2647F

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

Reference Manual

HEWLETT-PACKARD COMPANY
974 EAST ARQUES AVENUE, SUNNYVALE, CALIFORNIA, 94086

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

PREFACE

This manual is one of a series of manuals which documents use of the HP 2647F terminal. It consists of 13 sections, as follows:

SECTION	CONTENTS
1	*General Description.* Contains background information on the major physical and functional elements of the terminal.
2	*Display Control.* Describes how to control the terminal display from the keyboard and from a program.
3	*Terminal Control.* Describes how to control the terminal from a program, including configuration of the terminal, various keyboard operations, display enhancements, and selection of character sets.
4	*Graphics Control.* Describes use of the terminal's graphics capabilities from the keyboard and from a program.
5	*Commands.* Supplies a description, with examples, of each of the terminals commands: (Most of the terminals capabilities are implemented using commands.)
6	*Disc Operations From The Keyboard.* Describes how to perform terminal disc operations fromthe keyboard.
7	*Printer Operations From The Keyboard.* Describes how to control a printer connected to the terminal from the keyboard.
8	*Computer Operations.* Describes the keyboard operations required for data communications between the terminal and a computer.
9	*Data Communications.* Describes the physical connections and protocols used in terminal/computer data communications.
10	*Status.* Describes how to obtain and interpret the terminal's internal status, the status of a printer connected to the terminal, the status of any command sent to the terminal from a computer program, and the terminal's graphics status.
11	*Installation.* Contains installation procedures for the terminal itself and the terminals accessories and options.
12	*Configuration.* Provides configuration procedures for overall terminal operation and the printed circuit assemblies used in the terminal.
13	*Self Test.* Provides self test procedures for the terminal, the disc drive, and the datacomm circuits.
Appendix A	Error messages.
Appendix B	Terminal Specifications
Appendix C	Options and Accessories

Other related manuals are listed below:

1. *Getting Started with Your 2647F,* part no. 02647-90036 — This manual describes how to use the terminal for simple, everyday tasks.

2. *Forms/47,* part no. 13257-90020 — This manual describes how to enter data in forms and how to use the Forms Design mode feature to create forms.

3. *BASIC/47,* part no. 02647-90038 — This manual describes the BASIC commands and statements that are interpreted by the terminal BASIC interpreter.

4. *Autoplot/47,* part no. 02647-90042 — This manual describes how to make pie, bar, and linear charts by using the Multiplot application program and how to create overhead slide transparencies using the Slide application program.

5. *Word/47,* part no. 13257-90019 — This manual describes how to use the terminal for text preparation, creation of data entry forms, and how to enter data into the forms.

6. *2647F Service Manual Supplement,* part no. 02647-90043 — This supplement lists part numbers for replaceable parts, provides disassembly instructions, and test procedures for the terminal. (This manual is available at extra cost.)

7. *2647F Quick Reference Guide* — This guide lists the escape sequences required for entering a command from a program.

CONTENTS

ILLUSTRATIONS

GENERAL DESCRIPTION

SECTION 1

INTRODUCTION

The HP 2647F terminal is supplied with a 270K-byte mini disc drive with a second mini disc drive available as an option. Some of its features are: graphics, programmable in BASIC, text-editing, data entry forms design, and display enhancements. Control of the terminal itself, graphics, and terminal-controlled devices (such as discs, printers, and plotters) is enabled through keyboard keys and commands entered either at the keyboard or supplied as "escape sequences" by a program running on a computer.

The terminal uses a microprocessor under firmware control. The terminal provides interactive graphics features available under user or program control. It operates in character or block mode, with full editing capability. The terminal is designed for such applications as data entry and preparation, information display and editing, interactive programming, data communications, and time-sharing operation.

In addition, a mass storage capability of up to 270 kilobytes of data per diskette is provided. This allows the terminal to be used for either stand-alone or on-line operation. For example, forms designed at the terminal using the line drawing or other character sets can be stored on a diskette and selectively retrieved from the keyboard or through commands from a remote computer.

Data communications options are also available to provide a choice of communications capability. The standard terminal is teletypewriter compatible (EIA RS232-C serial asynchronous, ASCII, half or full duplex). It operates at speeds up to 9600 bits per second, and transmits either character-by-character as a fully interactive terminal or operates on variable length blocks of information. A 20mA current loop is optional. Also, the terminal can be used with a wide selection of modems over dialed or leased lines.

When the terminal is used with the HP 13265A 300 Baud Modem, auto-dialing is permitted (i.e., telephone numbers may be entered through the keyboard, then the modem dials the number). Also, telephone numbers may be stored in the terminal and called by user-assigned names.

A block diagram of the terminal is shown in figure 1-1. The terminal has four major and mechanically independent sections; keyboard, CRT monitor, mainframe, and mini disc drive. The specific functional properties of the terminal are determined by firmware programs resident in ROM (read-only-memory). It is these programs that make it possible for the terminal to have many powerful features such as self test, dynamic memory allocation, transparent control codes, and off-screen storage.

MAINFRAME

The heart of the system is the mainframe section, which can be considered a microcomputer system. In the mainframe is the power supply and a bus-oriented logic system containing the microprocessor, program and data memory, video display subsystem, keyboard interface, and data communications interface. The basic terminal contains five slots for options and accessories. All mainframe modules are functionally, mechanically, and electrically independent, giving a high degree of flexibility and reducing service time.

Microprocessor

The terminal uses an 8-bit microprocessor to control most of the terminal's operation. The microprocesor executes code that may be in ROM or RAM memories. It controls the display, programmable functions, and I/O devices.

Display

The display memory is used to hold up to 88 lines of 80 characters per line.

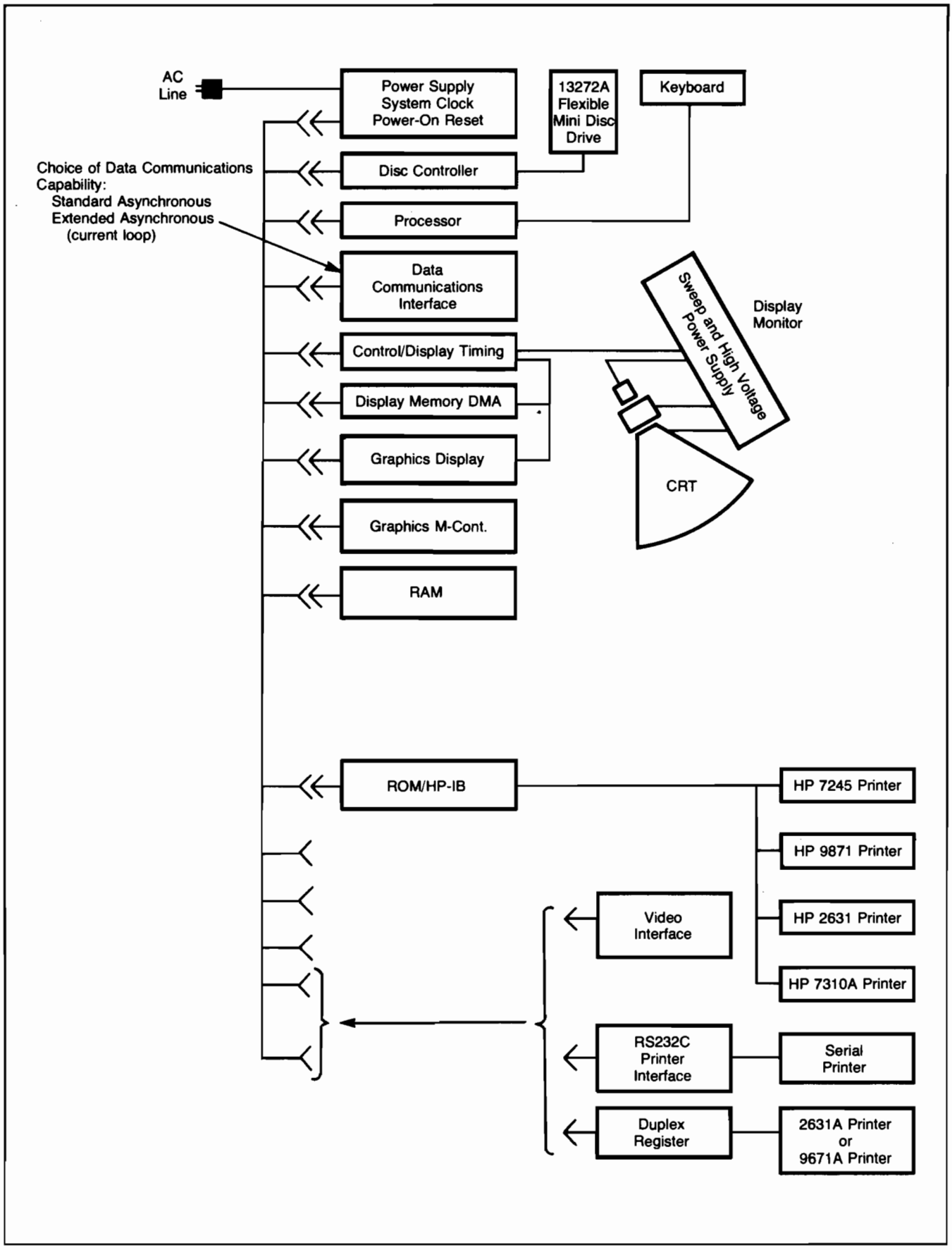


Figure 1-1. Terminal Architecture

Terminal Bus

A major element of the logic system is the terminal bus (a printed circuit board with connectors) which is attached to the bottom of the mainframe and to the power supply. The bus distributes power to the individual modules and provides data, address, and control lines for communication between the various logic functions. The terminal bus provides communication paths between processor, memory, input/output, and display refresh on a shared basis.

All modules are slot-independent and carry their own select code or memory address. The only requirements are that the display modules must be grouped together.

Terminal Memory

Like any other computer system, the microcomputer module is useful only if it has a program to execute and memory in which to store data. This is the function of the terminal memory modules, which are two types, read/write or random-access memory (RAM) and read-only memory (ROM). The RAM stores display characters and data; the ROM stores terminal programs (firmware). Terminal programs are called firmware because the ROM makes them more permanent than software but less permanent than hardware. In the terminal 96K of the available memory is dedicated to ROM or program memory and 128K of memory is RAM. All of the terminal memory is MOS semiconductor memory.

Input/Output PCAs

Also a part of the logic system are several terminal input/output modules: the Processor PCA, which handles keyboard input, the data communications PCA, the optional eight-bit duplex register PCA (used for the HP 9866A/B and HP 9871A Printer interface), the optional serial printer interface PCA, and the ROM/HP-IB PCA (used for interfacing the HP 9871A-001 Printer, HP2631B-046 Printer, HP 2631G Printer, HP 7245A-001 Plotter/Printer, HP 9872A Plotter, and other terminals). Note that plotting functions require a program running on a host computer to drive them.

These never request control of the bus, but all must respond to commands from the microcomputer and its programs.

The basic I/O commands output data or control codes from the microprocessor and input data or status from the interface module. Each of the I/O cards has different data and control formats, but all are controlled by the microcomputer. Each I/O module has a rear edge connector for the attachment of a connector hood and cable assembly to carry the signals out the back of the terminal.

CRT MONITOR

The CRT monitor section contains sweep, power supply, and high voltage circuits, the high-resolution, low-profile cathode-ray tube, and fan.

The Raster

The terminal uses raster scan deflection method, similar to that used in television sets. In a raster scan display, the electron beam traverses the screen in a series of closely spaced horizontal lines, starting from the top. Characters are formed from line segments and dots produced by turning the beam intensity on and off at appropriate times.

The terminal uses a low-profile CRT to keep overall height to a minimum while maintaining a screen capacity of 1920 characters, partitioned into 24 rows of 80 characters each. All of the character positions are fundamentally rectangles 7 dots wide by 9 scan lines high. Four additional scan lines beneath the 7×9 matrix are used for the descender areas of lowercase characters, for underlining, and for the blinking underscore cursor. One other dot is used on either side for character-to-character spacing, and one scan line is reserved at the top and bottom for row-to-row spacing. This results in a character cell of 9 dots by 15 scan lines replicated over the entire screen area (see figure 1-2).

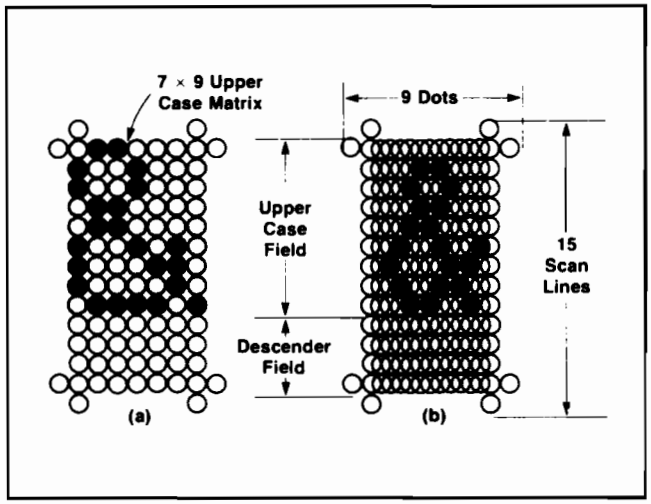


Figure 1-2. Character Cell

Alphanumeric and Microvector Character Sets

Two types of character sets can be stored within the terminal: alphanumeric sets and microvector sets. Alphanumeric sets support the primary use of the terminal, displaying textual and numeric information. Characters are designed around a basic 7×9 dot matrix with provision for lowercase descenders. The characters are embellished by use of the half-shift. With this type of set the character-to-character spacing of two dots is hardwired. This prevents the design of characters that would form continuous horizontal lines. However, all 15 scan lines of the two are available so that vertically contiguous symbol segments can be designed. An example of this is the three-row-high integral sign found in the math symbol set.

Microvector sets use the entire 9-dot-by-15-scan-line character cell without the half-shift. This allows characters to be designed with both horizontal and vertical continuity. This type of set finds its greatest application where a minimal set of graphic kernels is needed to represent more complex pictorial information. The data entry forms shown in Section III illustrate the use of the line drawing set in representing a form.

Display Features and Alternate Character Sets

The basic terminal uses four sets of 128 alphanumeric characters and four display features, inverse video (black characters on white background), half-bright, underline, and blinking fields.

All sixteen possible combinations of the four display features can be applied to any character or characters on the screen. No displayable character positions are required to start, stop, or modify either the features or the character sets. Therefore, consecutive characters on the screen may be from different sets or have different display features.

KEYBOARD

The processor scans the keyboard at discrete intervals for a depressed key. Each key is assigned a position in a matrix of 14 columns and 8 rows. This matrix provides a reference to a look-up table that the firmware uses to display the character and/or send the character code over the data communications line.

THE FIRMWARE

The firmware contains the operating system or main terminal code modules and the various input/output, data communications, and utility routines that control the terminal. The firmware is stored in read-only memory (ROM) circuits on the ROM/HP-IB PCA.

System Monitor

The system monitor is a section of the firmware that dispatches data within the terminal. The processor normally executes a basic loop, in which it scans the keyboard and the data communications interface and waits for something to happen (see figure 1-3). When a character is received from either the keyboard or the data communications interface, a general character interpretation routine is executed to determine the action to be taken. The monitor then performs the specified functions, such as putting a character on the display, transmitting a character over the data communications interface, or moving the cursor. When this has been completed, the monitor returns to the basic scan loop to look for the next input.

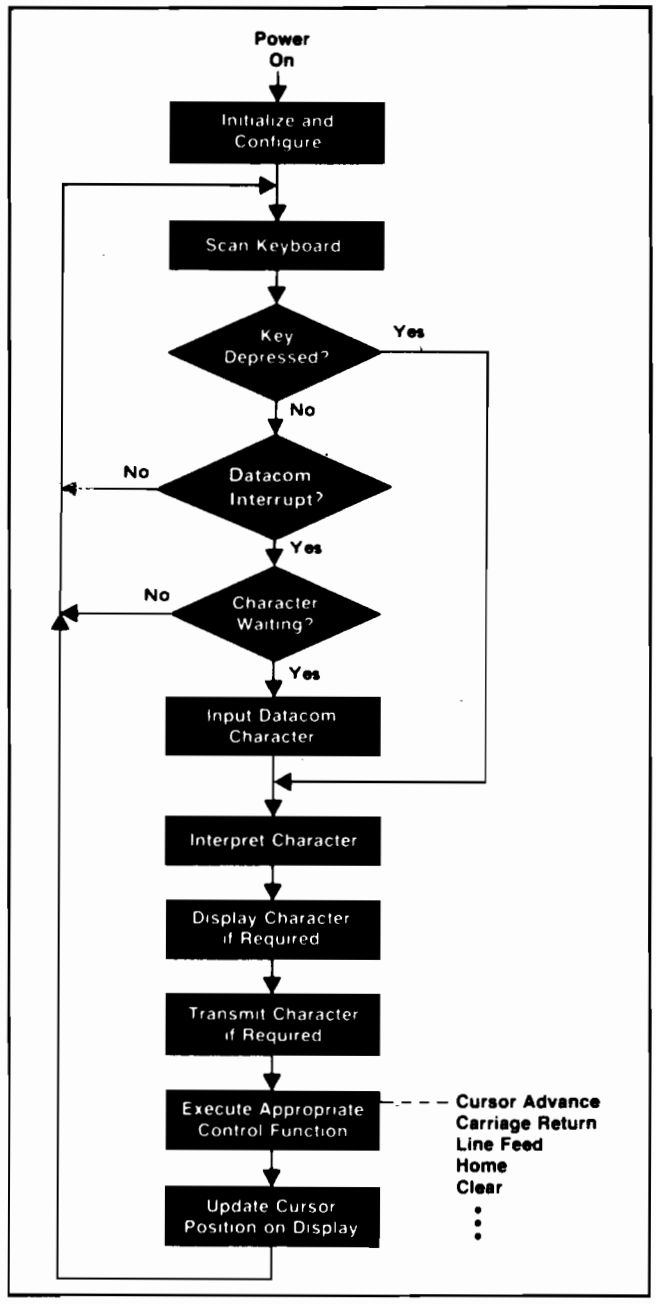


Figure 1-3. System Monitor Basic Loop

Keyboard and I/O Subsystem

The I/O subsystem contains the firmware required for performing all input/output functions. The firmware operates using both scan and interrupt methods. The keyboard is scanned at regular intervals, while inputs from the devices such as the data communications interface and cartridge tape units are interrupt driven. If a new key depression is detected, the key number associated with this key is calculated and used as an index into a table that assigns a code to the key. If the key is one of the ASCII keys, the proper code is determined based on the state of the CNTL, SHIFT, and CAPS LOCK keys.

If the key in question is not one of the ASCII keys, the firmware may be required to generate a multiple character sequence consisting of an ASCII escape character followed by one or more characters that define the escape sequence. Keys in a third group do not generate codes at all, but simply perform internal terminal functions, such as BLOCK MODE, REMOTE, and CAPS LOCK.

I/O associated with the display is minimal because the display memory access module (DMA) causes the display to be refreshed without processor intervention. Display I/O control mainly involves transmitting the cursor coordinates to the display whenever necessary.

Cursor Movement

The terminal firmware contains many subroutines for moving the cursor on the display. All cursor movement is handled by the firmware. When a character is typed on the keyboard and appears on the display, the cursor moves to the next column position because a cursor advance subroutine has been executed and has calculated a new cursor position. Similar subroutines exist for moving the cursor up, down, right, left, and home. The tab function is also a firmware routine; it uses a one-bit-per-column table to determine the next tab stop.

Display Memory Management

A large part of the terminal firmware is devoted to management of the display memory. Most conventional terminals use a byte of display memory for every displayable position on the CRT screen. If there are many short lines, as is frequently the case, there is a substantial amount of unused memory. The terminal does not allocate memory for character positions to the right of the last character entered, so this memory is available for other purposes. Turn on and turn off of the various display enhancements and character set selections, or start and end of unprotected fields between individual characters can be accomplished without an intermediate blank character position. With these features, the address of a character occupying a given row and column cannot be directly computed without some sort of scanning process.

The display memory consists basically of a linked list of fixed-sized blocks of RAM (see figure 1-4). This list is set up in such a way that the DMA can start at the first address on the screen and follow the list to produce an entire screen of information. All memory not currently allocated for display use is kept on a free-storage link list. Individual rows are linked with next and preceding rows, while blocks within a row are linked only in a forward direction. The storage allocated for a row may be as little as one block (16 bytes), or much larger than 80 characters, depending upon the number of displayable and non-displayable characters needed to create the row on the CRT.

The firmware finds the address corresponding to a given character position by starting at the last known position and moving through the list either backward or forward until it finds the new address. If the end of the list is found before the row in question has been found, blocks are removed from the free-storage list and used to create new rows. Once the correct row has been found, the firmware searches for the cursor column. If the end of the row is found before the column has been found, additional blocks are removed from the free-storage list and used to build the length of the row out to the column required. Whenever a block is required and free list is empty, an existing row must be released from display memory. This row is the first row of memory if the addition is at the end of memory, and is the last row of memory if a row other than the last row is being lengthened.

DATA COMMUNICATIONS

Data communications in the terminal is both a hardware and a firmware function. The data communications interface is a basic terminal module. This module has the necessary logic to interface the terminal bus to the communication line.

The communication interface accepts parallel data from the terminal, serializes it, and adds framing or synchronizing bits (start and stop). It performs the reverse process on incoming data, converting serial data to parallel and removing start and stop bits. The interface can generate and check parity and can also detect data overruns. A status word keeps the processor informed of the status of the interface.

The terminal firmware for communications has three main functions. First, the program reads the control settings on the keyboard, keyboard interface, and the data communications interface. Second, it processes input characters and transmits output characters. Many decisions are made on incoming characters, especially on control characters. The third function is modem control using control lines on the communications interface. Direct connections to a computer and Bell 103 type modems require a minimum of firmware control while Bell 202 type modems require more control. A more detailed description of the data communications operation is given in Section 8.

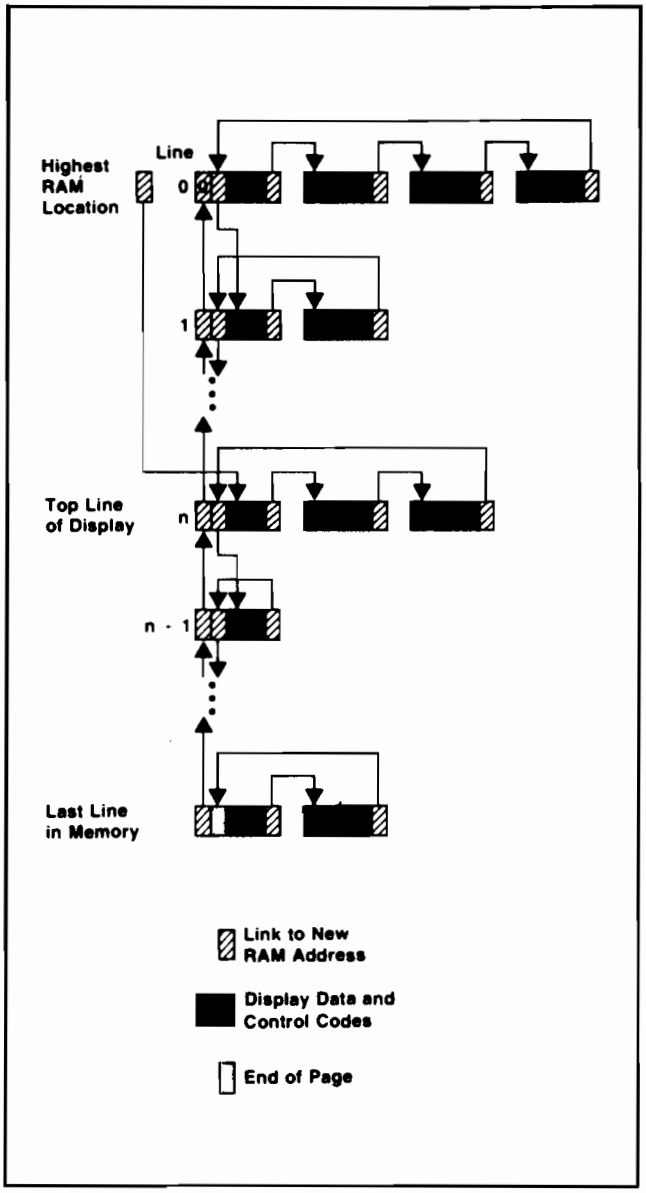


Figure 1-4. Display Memory Linked List Structure

THE FLEXIBLE MINI DISC DRIVE

The flexible mini disc drive operations are divided between hardware and firmware. The hardware consists of a disc drive controller, which is installed in the terminal, and up to two disc drives, which contain the disc drive electronics and electromechanical components. Power for disc drives is provided by the terminal power supply. The firmware is divided between the I/O code resident in ROM on the ROM/HP-IB PCA and disc controller code resident in ROM on the Disc Drive Controller PCA.

The controller hardware (which consists of a processor chip, RAM, ROM, disc controller chip) provides the interfacing between the terminal and the disc drive.

The firmware translates the disc commands from the command channel into commands recognized by the disc controller chip on the disc controller PCA. The disc controller chip, in turn, provides the signals for controlling the disc drives.

THE DISKETTE

Each diskette is a 5-1/4 inch (133 millimeter) flexible plastic disc enclosed in a plastic jacket. The surface of the disc is coated with ferromagnetic iron oxide. Data is stored in the form of binary digits represented by magnetic flux reversals on the disc. Information is stored and retrieved by the disc unit's heads that come in contact with the disc's upper and lower surfaces.

The diskette is double-sided and double-density which means that data is written on both sides of the disc and that 270K bytes (characters) of formatted information can be stored. The diskette is LIF compatible which allows the diskettes to be used between different HP systems (i.e., data written on the diskette from one HP system can be read by another HP system).

Information is stored on 33 concentric tracks on each side of the disc. Each track is divided into 16 sectors. Each sector is a *fixed length physical record* of 256 bytes (characters). The 1056 sectors are functionally divided into three major areas:

1. the volume label area (sectors 0 and 1),
2. the directory area (sectors 2 through 31), and
3. the data area (sectors 32 through 1055).

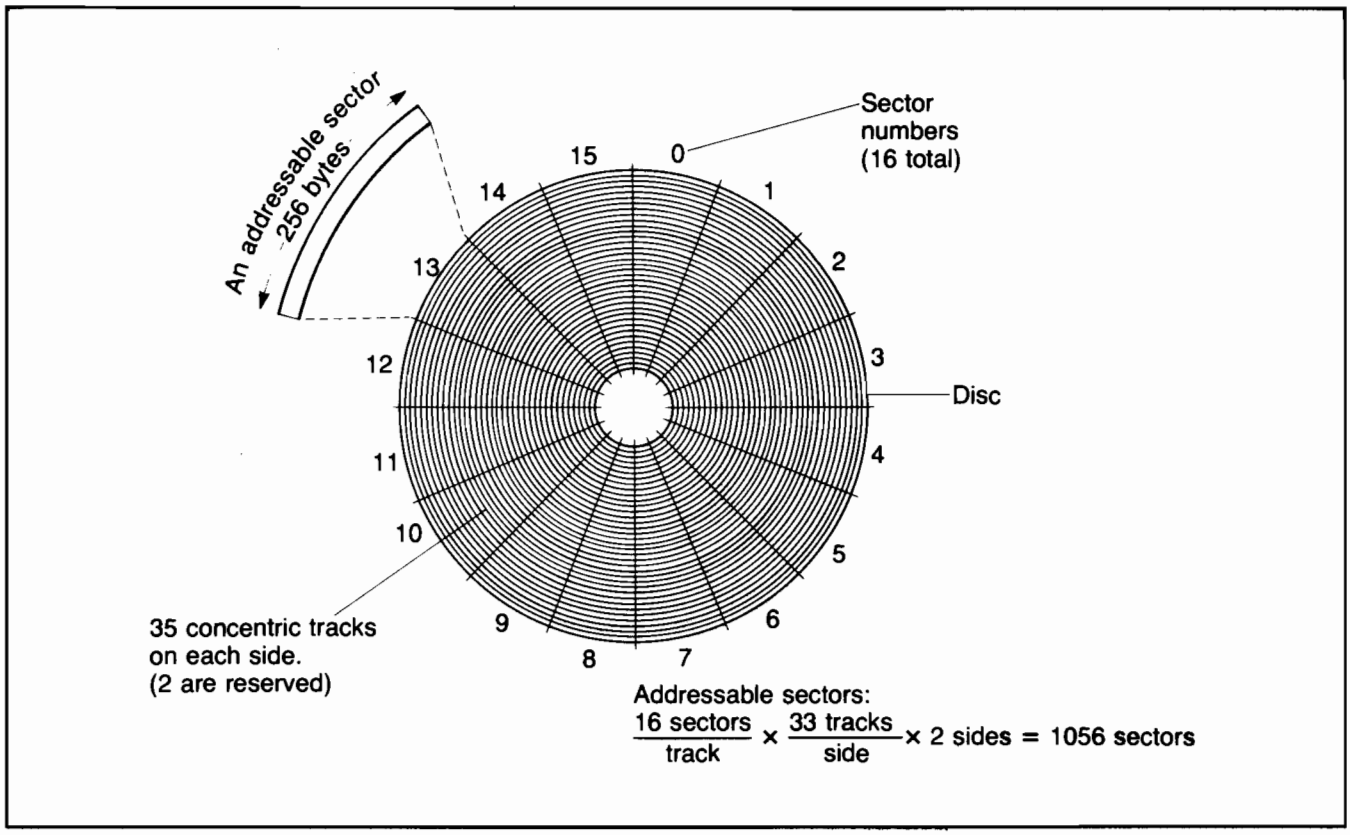


Figure 1-5. Diskette Particulars

The Volume Label Area. The volume label consists of:

1. the header bytes,
2. volume name,
3. directory start address,
4. length of the current directory in sectors, and
5. the maximum tracks, heads, and sectors for the diskette.

```
VOLUME NAME        HP-IB ADDRESS
-----------------  ------------------

Local   Disc       Start of    Space  Sector   Create        Create    Write
Volume  Type#Unit  Free Space  Left   Stagger  Date (M/D/Y)  Time      Protect
------  ---------  ----------  -----  -------  ------------  --------  -------

USER    mfD#1           199    857        9   9/30/81       11:40 AM  NO
```

A Sample Show Volumes Listing

The Directory Area. The directory area contains the following data for each file on the diskette:

1. file name
2. file type (ASCII or binary)
3. starting sector address of the file
4. length of the file in sectors
5. creation date of the file
6. volume number

```
VOLUME = USER

File        File   Start    Size        Create        Create
Name        Type   Address  (sectors)   Date (M/D/Y)  Time
----------  -----  -------  ----------  ------------  --------

FORECAST1   Ascii      32          5     9/23/80     9:23 AM
FORECAST7   Ascii      38          8     9/29/80     2:04 PM
```

A Sample Show Files Listing

The directory allows random access to the files by filename. Whenever a file is accessed, the directory is read to find the starting address of the file.

The Data Area. The data area for each file is structured as:

1. record length
2. data record
3. EOF
   .
   .
   .
4. End-of-File mark

Data is stored on the diskettes in *variable length logical records* (a line of data from the display will be stored as a logical record); a logical record may cross over sector

boundaries. All new files are started at the beginning of a sector. Up to 255 bytes may be unused between the end-of-data mark in one sector to the beginning of the next file in the following sector. The length of each logical record (a line of characters) written on the diskette is the number of characters (bytes) in the line, plus two bytes of overhead. Two bytes of the overhead define the length of the record, and the other two bytes terminate the records (a [illegible]).

DISPLAY CONTROL

SECTION 2

INTRODUCTION

This section contains information for using the display memory functions. The display memory functions change the position of display data or assign special attributes to blocks or fields of display data within each display workspace. The special attributes alter the way data is displayed or transmitted.

DISPLAY CONTROL

Cursor and display positioning operations can be programmed within each of four display workspaces. The programmable functions available are as follows:

- Display Window Control
  - Message Line
  - Command Line
  - Soft Key Label Line
- Cursor Sensing
  - Absolute
  - Relative
- Cursor Positioning
  - Absolute Addressing
  - Screen Relative Addressing
  - Cursor Relative Addressing
  - Space
  - Backspace
  - Set Tab
  - Clear Tab
  - Tab
  - Backtab
  - Set Margins
  - Home Up
  - Home Down
- Display Positioning
  - Roll Up
  - Roll Down
  - Next Page
  - Previous Page
  - Display Lock (Memory Lock)

Display Windows

Alphanumeric display memory is shared by four dynamic display workspaces (window#1 through #4), a message line (window#5), a command line (window#6), and a soft key label line (window#7). These "windows" are controlled through the command channel (Figure 2-1 shows the display features.)

~t,cDIsplay Window#<n>~	displays the selected window
~t,cCLOse Window#<n>~	removes selected window #5, #6, or #7

Display Workspaces

Only one display workspace can be displayed at a time. Cursor and display positioning operations operate independently in each workspace (i.e., you can roll up, roll down, next page, previous page, set margins and tabs, and position the cursor independently in each workspace.) As data is entered into a display workspace, it uses some of the alphanumeric display memory. If needed, the present display workspace will "steal" lines from the other workspaces that are not memory locked. Therefore, to preserve information in another workspace, it must be memory locked. At power on or full reset, windows#2 through #4 are memory locked. (Figure 2-2 shows examples of memory allocation for the four display memory workspaces.)

Message Line

The message line (window#5) is automatically displayed when an incorrect operation is performed. All messages that the terminal generates are given in the appendix. You may use the message window in a program to instruct the user to perform an operation (such as, insert a disc, or enter some data from the keyboard) by sending:

~t,cDIsplay Window#5~ <message string>~

To remove the message:

~t,cCLOse Window#5~

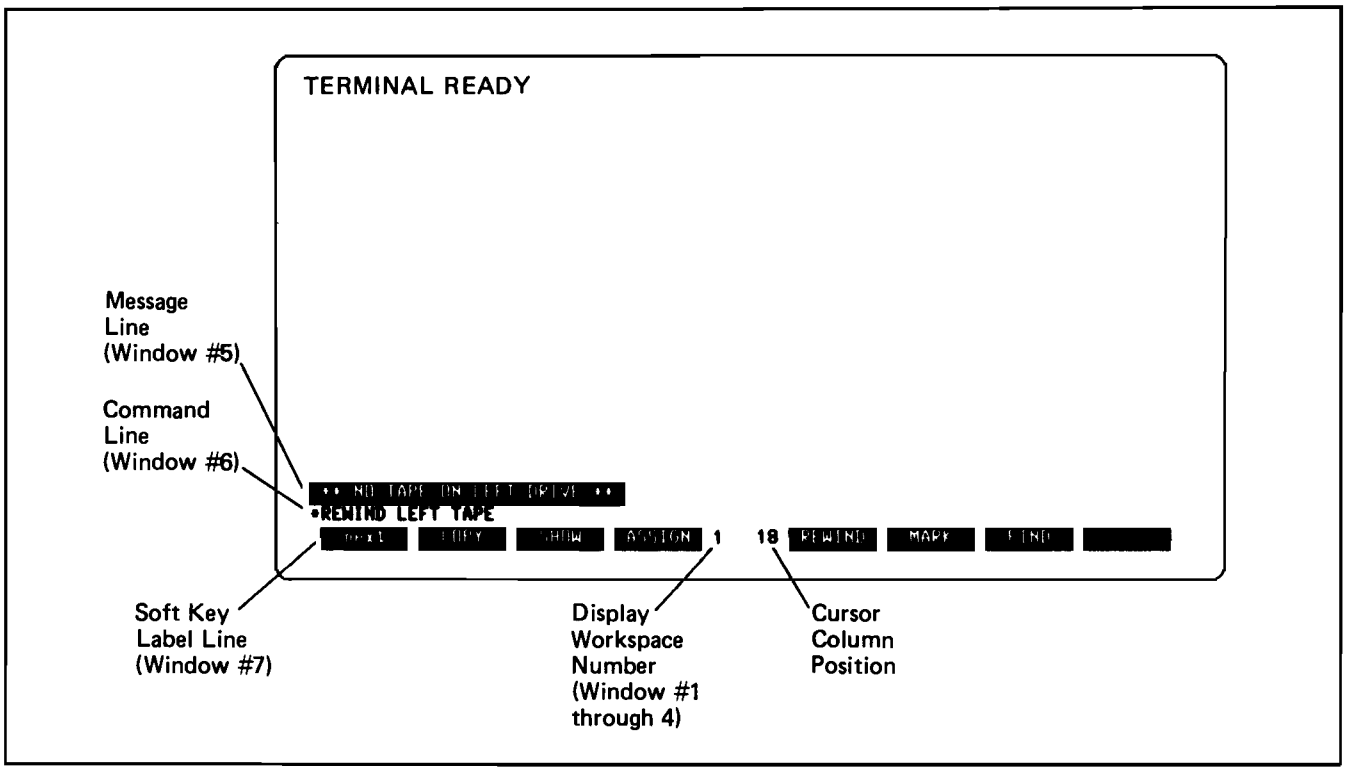


Figure 2-1. Display Features

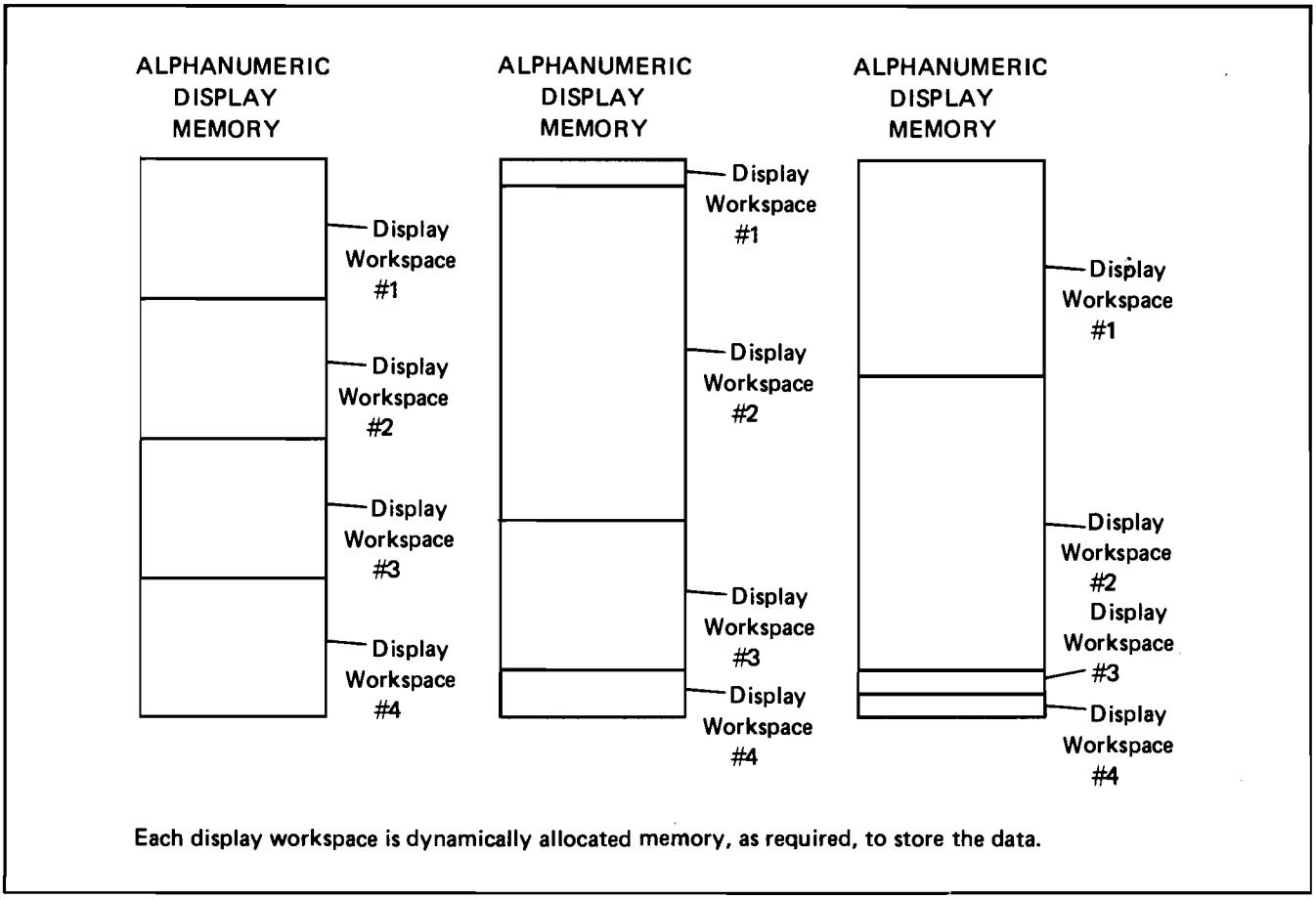


Figure 2-2. Three Examples of Display Memory Allocation to the Four Display Workspaces

Command Line

The command line is displayed to allow the user to enter commands into the command channel while under program control.

```
%,cDIsplay Window#6%
```

To remove the command line:

```
%,cCLOse Window#6%
```

Soft Key Label Line

The soft key label line displays either the command soft key labels (displayed when the [COMMAND] key is pressed) or the user-defined function labels for the soft keys (f1 through f8). Predefined user functions may be loaded into the soft keys under program control (see "Defining Soft Keys" explained later in this section). The labels may then be displayed to allow the user to select one of the predefined functions while running under program control:

```
%,cDIsplay Window#7%
```

or

```
%&jB
```

To remove the soft key label line:

```
%,cCLOse Window#7%
```

or

```
%&j@
```

MEMORY ADDRESS SCHEME

Display workspace positions can be addressed using absolute or relative coordinate values. Each display workspace is made up of 80 columns (0-79) and a number of rows determined by the size of the workspace. There can be as many as 97 lines of 80 characters (3 screens). The types of addressing available are:

- Absolute
- Screen Relative
- Cursor Relative

Row Addressing

Figure 2-3 illustrates the way the three types of addressing affect row or line numbers. The cursor is shown positioned in the fourth row on the screen. Screen row 0 is currently at row 6 of display memory. To reposition the cursor to the first line of the screen the following three destination rows could be used:

a. Absooolute: row 6
b. Screen Relative: row 0
c. Cursor Relative: row −3

Column Addressing

Column addressing is accomplished in a manner similar to row addressing. There is no difference between screen and absolute column addressing. Figure 2-4 illustrates the difference between absolute and relative addressing. The cursor is shown in column 5.

Whenever the row or column addresses exceed those available, the largest possible value is substituted. In screen relative addressing, the cursor cannot be moved to a row position that is not currently displayed. For example, in figure 2-3c a relative row address of −10 would cause the cursor to be positioned at the top of the current screen (relative row −3). Column positions are limited to the available screen positions (0 to 79 in figure 2-4a and −5 to 74 in figure 2-4b). The cursor cannot be wrapped around from column 0 to column 79 by specifying large negative values for relative column positions.

Note the difference between this type of addressing (0-79) and the column indicator in the soft key label line (1-80).

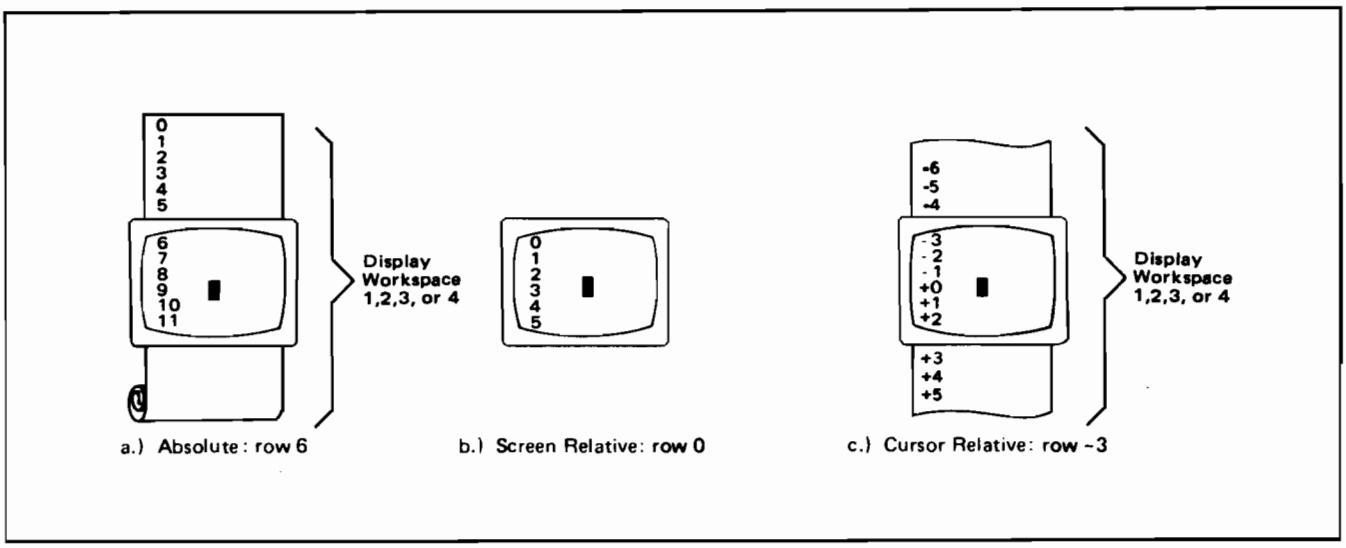


Figure 2-3. Row Addressing

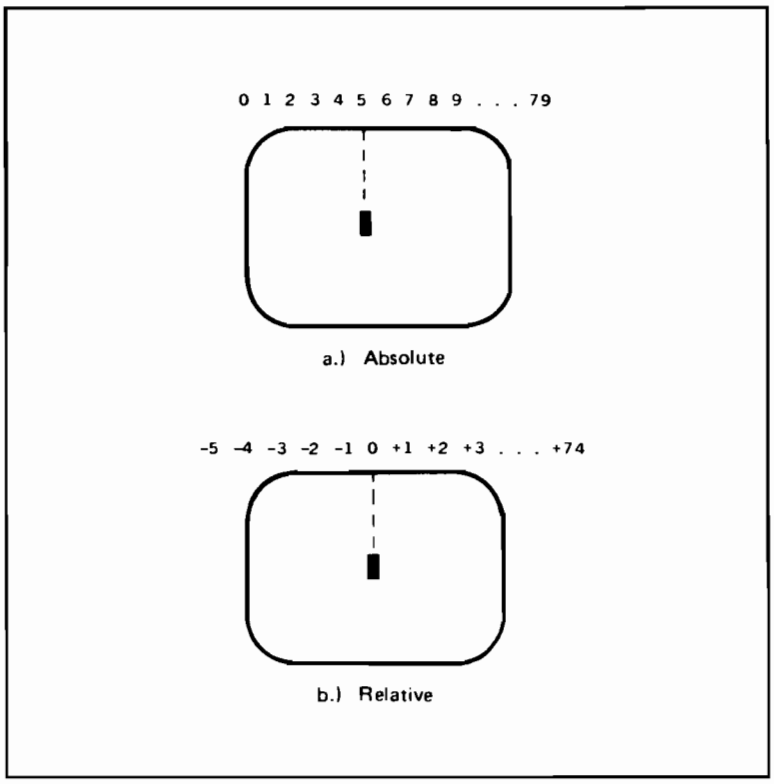


Figure 2-4. Column Addressing

CURSOR SENSING

The current position of the screen cursor can be sensed. The position returned can be the absolute position in display memory or the location relative to the current screen position. (Absolute and relative addresses are discussed under Memory Addressing.)

Absolute Sensing:

Example: The cursor is at column 20, row 40.

computer: ^[a
terminal: ^[&a020c040R

Relative Sensing: ^[`

Example: The cursor is again at column 20, row 40, but screen row 9 begins at row 35 of display memory.

computer: ^[`
terminal: ^[&a020c005Y

CURSOR POSITIONING

The cursor can be positioned directly by giving memory or screen coordinates, or by sending the escape codes for any of the keyboard cursor positioning operations.

Absolute Addressing

The cursor can be positioned to any displayable position using absolute coordinates. Absolute cursor positioning is accomplished using the following sequence:

^[&a <row number> r <column number> C

where: row number is 0 to 255.

Example: Position the cursor at row 35, column 6.

^[&a35r6C
or
^[&a6c35R

Absolute addressing cannot be used while Memory Lock is on.

Screen Relative Addressing

The cursor can be positioned to any position currently displayed on the screen by using screen relative co-ordinates:

^[&a <screen row number> y <column number> C

where: the top row of the screen is now 0.

Example: Position the cursor to screen relative row 15, column 53.

^[&a15y53C
or
^[&a53c15Y

Cursor Relative Addressing

The cursor can be positioned to any displayable position by using cursor relative coordinates. (±row, ±column). Cursor relative addressing cannot be used while Memory Lock is on.

Example: The cursor is currently at row 7 and column 10 of the screen. Move the cursor to row 9 column 6.

```
␛&a+2r-4C
    or
␛&a-4c+2R
```

Combinations of Absolute and Relative Addressing

Relative and absolute coordinates can be combined in the same sequence.

Example: Move cursor from its current row down 8 rows and to column 60.

```
␛&a+8r60C
    or
␛&a60c+8R
```

Example: Move cursor from its current position 15 columns left, and to relative screen row 4.

```
␛&a4y-15C
    or
␛&a-15c4Y
```

OTHER CURSOR OPERATIONS

In addition to positioning the cursor using coordinates, you can use a variety of keyboard equivalent operations. These operations normally require only one or two characters to be sent to the terminal. Table 2-1 lists each of the operations together with its code and a brief description.

Tabs

You can programmatically set tabs, tab, and clear tabs independently in each of the workspaces.

Setting Tabs. To set a tab, move the cursor to the desired column and send ␛1. Once a tab is set, the tab function or [TAB] key can be used to move the cursor to the next tab setting. In Forms Mode, previously set tabs are ignored; however, when Forms Mode is turned off, the previously set tabs are in effect.

Using Tabs. Once tab positions have been set you can tab in the same manner that you would on a typewriter. You can even tab backwards to the previous tab position by sending ␛i. When you are at the first tab position in a line and you backtab, the cursor moves to the last tab position in the previous line. Once the cursor has reached the first tab position in the first line of memory, no further backtabbing movement can be made.

Clearing Tabs. You can clear individual tabs by moving the cursor to the tab position and sending ␛2. All of the tabs in a workspace can be cleared at once without having to position the cursor by sending ␛3.

Margins

You can set the left and right margins in each workspace to make the entry of data easier. When the terminal is turned on or a full reset performed, the margins are set at columns 0 and 79 in each of the workspaces. This gives a full 80 character line. You can define new margins as follows:

Left Margin. Move the cursor to the desired left margin setting. Send ␛4.

Set Left Margin: ␛4

Right Margin: Move the cursor to the desired right margin setting. Send ␛5.

Set Right Margin: ␛5

The terminal will beep when you are eight characters from the right margin. When the right margin is reached, the cursor will move to the left margin of the next line.

Example: Set the margins for a 40 column page centered on the screen.

Move the cursor to column 20 and set the left margin. Move the cursor to column 59 and set the right margin. Place the cursor back at column 20 and begin sending data.

```
␛&a20C             ␛4       ␛&a59C            ␛5
\      /          \   /\          /       \  /
position cursor   set margin   position cursor   set
                                                 margin
                  column numbers
  2           3           4           5           6
  0           0           0           0           0

          This is an example using margins to
                  control data entry.
```

Margins are cleared or changed by setting new margins (or a full reset) or by entering forms mode, where the margins are reset to columns 0 and 79.

Table 2-1. Cursor/Display Operations

FUNCTION	CODE	DESCRIPTION
Cursor		
	ESC*dq	Turn cursor on.
	ESC*dr	Turn cursor off.
Line Feed	LF (J^c)	Move the cursor to the next line.
Return	CR (M^c)	Return the cursor to the left margin, halt I/O operations, and clear messages.
	ESC G	Move cursor to first column of current row.
Backspace	BS (H^c)	Move the cursor one column to the left. If the cursor is in column 0, it remains there.
Cursor Up	ESC A	Move the cursor up one row. If the cursor is in row 0, it wraps around to row 23.
Cursor Down	ESC B	Move the cursor down one row. If the cursor is in row 23, it wraps around to row 0.
Cursor Right	ESC C	Move the cursor right one column. If the cursor is in column 79, it wraps around to column 0 of the next row. If the cursor is in row 23, column 79, it wraps around to row 0, column 0.
Cursor Left	ESC D	Move the cursor left one column. If the cursor is in column 0, it wraps around to column 79 of the previous row. If the cursor is in row 0, column 0, it wraps around to row 23, column 79.
Home Up	ESC h	Move the cursor to the beginning of the first line of the current display workspace (excluding transmit-only fields in Format Mode.)
Home Up	ESC H	Move cursor to the beginning of the first line in the current display workspace (including transmit-only fields in Format Mode).
Home Down	ESC F	Move the cursor to the beginning of the line following the last data in the current display workspace.
Tabs		
Tab	HT (I^c) or ESC I	Move the cursor forward to the next tab position.
Back Tab	ESC i	Move the cursor back to the previous tab position.
Set Tab	ESC 1	Place a tab at the current cursor column.
Clear Tab	ESC 2	Clear the tab at the current cursor column.
Clear All Tabs	ESC 3	Clear all tabs in the current display workspace.

FUNCTION	CODE	DESCRIPTION
Margins		
Set Left	ESC 4	Set the left margin at the current cursor column.
Set Right	ESC 5	Set the right margin at the current cursor column.
Display		
	ESC*de	Turn on current alphanumeric display workspace.
	ESC*df	Turn off current alphanumeric display workspace.
Select Display Workspace, Message, Command, or Soft Key Label Line	ESC,c DIsplay Window# <n> CR	Selects display workspace n (where n is 1, 2, 3, or 4), message line (n×5), command line (n×6), or soft key label line (n×7).
Close Window 5, 6, or 7	ESC,c CLOse Window # <n> CR	Removes message, command, or soft key label line from display.
Clear Display	ESC J	Clear the current display workspace from the cursor position to the end of the workspace.
Clear to End of Line	ESC K	Clear current line beginning at the column containing the cursor.
Roll Up	ESC S	Roll the screen up one row (until the last row of memory is located at the top of the display). Cursor is stationary.
Roll Down	ESC T	Roll the screen down one row (until the first row of memory is located at the top of the display). Cursor is stationary.
Next Page	ESC U	Display the next 24 rows of the current display workspace (until the last row of the current display workspace is located at the top of the display.) The cursor is moved to the first unprotected location on the new page when in forms mode.
Prev Page	ESC V	Displays the previous 24 rows of the current display workspace (until the first row of the current display workspace is located at the top of the display). The cursor is moved to the first unprotected location on the new page when in forms mode.
Memory Lock	ESC l	Turn on memory lock (overflow protect). Note that when Memory Lock is on, only screen relative addressing can be used.
	ESC m	Turn off memory lock. Refer to the *User Manual* for additional information on Memory Lock.

TERMINAL CONTROL

SECTION 3

TERMINAL CONTROL FUNCTIONS

The following paragraphs describe how to programmatically change most of the terminal's control settings and perform various other control functions. The settings consist of the following:

- Latching Keys [AUTO LF], [BLOCK MODE], [CAPS LOCK], and [REMOTE].
- Configuration Switching Settings (A-Z)

When the terminal is powered on or reset, the states of the various keyboard and internal switches are stored in memory. Most of these stored settings can be changed programmatically without physically changing the switch setting. (Note that if the terminal is reset the terminal will return to the physical setting.)

You can select a specific operating configuration from within the application program. This eliminates the problem of requiring the terminal operator to make the settings before continuing with the application program. It also allows individual programs or even subroutines to change terminal configuration for a specific function and then return the terminal to the original state before passing control back to the main program.

Latching Keys

Four of the mechanically latching keys, [AUTO LF], [BLOCK MODE], [CAPS LOCK], and [REMOTE] can have their electronic state changed programmatically. The escape code sequence is as follows:

```
ESC & k <state> <key> . . .
```

where:

<state> is
- 0 (up)
- or
- 1 (down)

<key> is
- a (AUTO LF)
- or
- b (BLOCK MODE)
- or
- c (CAPS LOCK)
- or
- r (REMOTE)

The ESC & k is followed by one or more groups of state and key parameters. The state is a 0 or 1 to indicate that the key is to be up or down respectively. The key is a single letter a, b, c, or r. The groups can be in any order ([CAPS LOCK] can come before [BLOCK MODE]). The last key letter in the sequence must be capitalized to indicate the end of the sequence.

Example: Set [AUTO LF] up and [CAPS LOCK] down.

```
ESC & k 0 a 1 C
```

An invalid character (any character other than 0, 1, a, b, c, or r) will cause the entire sequence to be ignored. An improper setting will cause only the invalid setting to be ignored. The rest of the sequence will be accepted.

Configuration Switches

The configuration switches on the Keyboard Interface PCA allow you to alter terminal operation for specific applications. Table 3-1 contains a summary of the switches and their function. A more complete description of the switches is given in the Installation section.

The switch settings are made using the following sequence:

```
ESC & s <state> <switch> . . .
```

where:

<state> is
- 0 (closed)
- or
- 1 (open)

<switch> is
- A through Z less I and O

An invalid character in the sequence will cause the tire sequence to be ignored. The sequence must be terminated with an upper case switch character. A full reset will cause the terminal to return to the physical settings of the switches. Switches S and T cannot be changed if the terminal is configured for Main Channel protocol.

Example: Set switches A, B, and D open and switch C closed.

```
ESC & s 1 a 1 b 1 d 0 C
```

In certain operating configurations (i.e., multipoint), some switch settings cannot be changed. If attempted, the new setting for the switch will be ignored.

Table 3-1. Keyboard Interface Switch Summary

SWITCH	POINT-TO-POINT FUNCTION
A	Function key transmission
B	Space overwrite latch
C	Cursor end-of-line wraparound
D	Block mode (Line/Page)
E	Paper tape mode
F	Fast binary read
G	Block transfer handshake
H	Inhibit DC2
J	Auto terminate
K	Clear terminator
L	Self-test inhibit
M	Reverse CNTL key effect on INSERT CHAR and DELETE CHAR keys
N	Escape code transfer to printer
P	Compatibility Mode (scaled)
Q	Compatibility Mode (unscaled)
R	Circuit Assurance
S	Main Channel Protocol
T	Main Channel Protocol
U	CPU break
V	Carrier Detect
W	DC1/DC3 Handshake
X	Data speed select
Y	Transmit LED
Z	Parity

ADDITIONAL CONTROL FUNCTIONS

In addition to the control settings there are several control operations that can be controlled programmatically. These control functions are as follows:

- Bell - G^c
- Send Display - ↑t d
- Wait - ↑t @
- Keyboard Disable - ↑t c
- Keyboard Enable - ↑t b
- Reset Terminal (Soft) - ↑t g
- Reset Terminal (Full) - ↑t E
- Terminal Self-Test - ↑t z
- Data Comm Self-Test - ↑t x
- Modem Disconnect - ↑t f
- Program Down Loading - ↑t & b or ↑t & c
- Display message, command, or soft key label line - ↑t, c <command sequence>↑k

Bell

The G^c character causes the terminal to "beep". A beep is automatically generated at the end of each unprotected field in format mode and during normal operation as the cursor passes within eight positions of the right margin.

Send Display

The ↑t d sequence causes the terminal to send a block of the current workspace data to the computer. The data sent depends on the Line/Page setting of Keyboard Interface switch D and whether the terminal is in format mode or not.

Data is transmitted beginning at the current cursor position. If the terminal is strapped for page, data is transmitted until the end of the current display. If strapped for line, transmission stops at the end of the current line for non format mode or at the end of the current field if in format mode.

Wait

The terminal can be made to pause for approximately 1 second by sending it an ↑t @. Multiple commands can be used to obtain any desired time period.

Keyboard Disable/Enable

The terminal keyboard can be locked by sending an ↑t c. It must then be unlocked by sending an ↑t b or by pressing the RESET TERMINAL key.

Reset Terminal

A programmatic "Soft Reset" can be made by sending ↑t g to the terminal. A "Full Reset" can be made using ↑t E.

Soft Reset (↑t g). A soft reset results in the following:

1. Any error messages present are cleared, the normal display is returned and the keyboard is unlocked.
2. If DISPLAY FUNCTIONS is enabled, it is turned off.
3. If the terminal is set for REMOTE, RECORD operations are ended.
4. Printer operations are stopped.
5. Current transmission of data stops. Data waiting to be sent to the computer is not sent. Partial messages from the computer are lost. The data communications facility is re-initialized.
6. All keyboard lights are turned on for 0.5 seconds.

Full Reset (ESC E). A full reset has the same effect as turning power on and consists of the following:

The screen and memory are cleared, then TERMINAL READY is displayed in workspace 1. Format mode, display functions, and all programmable functions including the function keys (f0-f8) are turned off or set to their default values.

NOTE: The CPU must wait five seconds after issuing ESC E before sending additional data.

Self-Test

The Terminal Self-Test can be executed by sending ESC z. The Data Comm Self-Test can be executed by sending ESC x. Descriptions of the self-tests are given in Section IX.

Modem Disconnect

The terminal can be directed to "hang up" the modem by sending an ESC f. The terminal does this by lowering the CD (Data Terminal Ready) line for 1 second if 13260A or 13260B data comm is used or 10 seconds if the terminal is configured for multipoint.

Program Down Load

The ESC & b and ESC & c sequences allow special diagnostic programs to be loaded into the terminal and executed. The escape sequence must precede the program to be loaded. This function can be used by HP diagnostics only. The ESC & c functions the same as ESC & b except that the LOADER message is not displayed.

Display Message, Command, or Soft Key Label Line

Any or all of the message, command, or soft key label lines may be displayed by sending ESC, c DIsplay Window#<n>%.

where n 5 - message line
6 - command line
7 - soft key label line

DISPLAY ENHANCEMENTS

The standard terminal can display data using inverse video (black on white), half bright, underline, and blinking characters. Each character position on the screen can be displayed with various combinations of these features.

- **Half Bright** — characters are displayed at half intensity (grey).
- **Underline** — an underline is displayed below the normal character.
- **Inverse Video** — the screen is white and characters are black.
- **Blinking** — characters including the inverse video, underline, and half bright features blink.

The display enhancements are used by assigning one or more of them to a field. The selection sequence is:

ESC & d <enhancement character>

The enhancement character (@, A through O) is used to select the combination of display enhancements to be assigned to the field. The following table lists the enhancement character for each of the combinations. The field is ended by selecting another enhancement, the end of the current line, or by ESC & d @.

	Enhancement Character														
	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Half Bright									x	x	x	x	x	x	x	x
Under line					x	x	x	x					x	x	x	x
Inverse Video			x	x			x	x			x	x			x	x
Blinking		x		x		x		x		x		x		x		x
End Enhancement	x															

Example: Define columns 10 through 14 of line 5 to be inverse video and blinking.

Step 1. Position the cursor at column 10 in line 5.

Step 2. Send ESC & d C.

Step 3. Move the cursor to column 15 in line 5.

Step 4. Send ESC & d @ (this ends the enhancements). The field should be white.

Step 5. Send the word TERMINAL beginning in column 9 of line 5. It should appear as shown below.

ALTERNATE CHARACTER SETS

The terminal can display up to four different character sets. Each character set can contain up to 128 characters or symbols.

Switching from one character set to another can be done on a character-by-character basis. For example, a character from the Math Symbol set can be displayed next to characters from the Roman set. This is done by defining one or more character positions in a line to be from alternate character sets. (Each group of characters can be thought of as a field.)

NOTE: The following discussion assumes that the Math and Line Drawing character sets are present and are installed as alternate sets A and B respectively, which is the standard installation.

Selecting Alternate Sets

To use optional character sets, first select the character set to be used as the alternate. (With the terminal in its initial state, character set A is defined to be the alternate.) An alternate set is selected with the following sequence:

ESC) <set>

where: set = @, A, B, or C

Note that if @ is used, the Roman or basic terminal set would be selected as the alternate. To find out which character set corresponds to @, A, B, or C, generate the test pattern by pressing the TEST key). This displays the order of the character sets as shown.

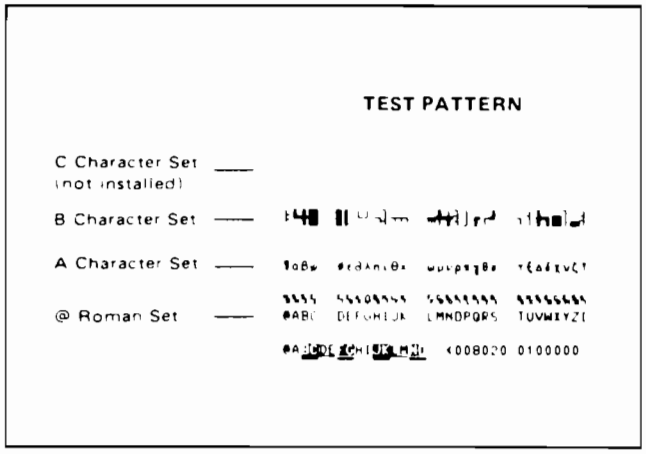


Figure 3-1. Character Set Locations

Using Alternate Sets.

Once the alternate character set is defined, you can switch from the Roman to the alternate set with N^c (SO).

The terminal automatically returns to the base or Roman set at the end of a line. To return to the base set within a line, send a O^c (SI). This means that you must send another N^c to turn on the alternate set if it extends to the next line.

Example: Define the Math Set as the alternate character set.

From the test pattern the Math Symbol Set is found to be the A alternate character set.

An alternate set is selected with the following sequence:

ESC) A

To display AαBβ send the following sequence:

A N^c A O^c B N^c B

The screen should display AαBβ:

AαBβ

Once an alternate set field has been defined, the field moves with the display (i.e., roll up, roll down, etc.). To change to a different alternate character set another ESC) <set> sequence must be sent.

Once a field in display memory has been defined as an alternate character field, it will continue to display alternate characters whenever data is written in the field until the terminal is reset or the portion of the line containing the alternate character set control code (N^c) is deleted.

The Math Set is useful for applications requiring the use of equations or formulas. The elements of the Math Symbol Set are shown in figure 3-4. An example of the use of the Math Set is shown in figure 3-2.

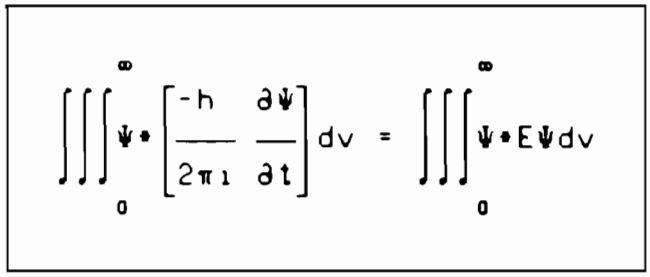


Figure 3-2. Example Using the Math Set

The Large Character Set allows you to create alphabetic characters that are three times the size of normal characters. The elements of the Large Character Set are shown in figure 3-5. An example of how to use the Large Character Set to build the character "B" is shown in figure 3-3. The keys required to build each character and a program to build each character are given in the appendix.

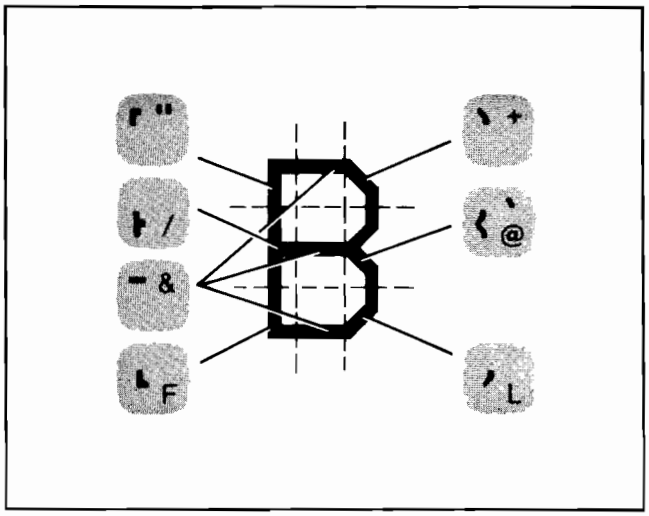


Figure 3-3. Example Using the Large Character Set

The Line Drawing Set provides a limited graphics capability. Simple line drawings and fairly complex forms for data entry applications can be generated. The elements of the Line Drawing Set are shown in figure 3-6. Figure 3-7 shows how the line drawing set can be used to build a data entry form.

PROGRAMMABLE SOFT KEYS

The terminal has 8 programmable keys [f1] through [f8]. In addition, the [RETURN] key can also be assigned a string value. The [RETURN] key is addressed as the f0 key in escape sequences. All 9 keys can be used by the operator or triggered from a program. Each key can be assigned a label of up to 8 characters and a string of up to 80 characters. The keys can be defined to be used at the terminal only (L), transmitted to the computer only (T), or to be treated as normal keyboard input (N). The keys can be programmed with escape code sequences to control or modify terminal operation. The keys can be used in application programs to create "menu" lists of special commands or in the case of [RETURN] to create a terminator for communications protocol.

Controlling the Soft Key Menu

You can cause the current soft key assignments to be displayed using the following escape sequence:

Display soft keys menu: ESC j

This will also allow the terminal operator to enter new key assignments from the keyboard. Procedures for entering new soft key assignments by escape sequences are given next in this section. The soft key display is in the following format:

```
F # type label
string
```

where: "#" is the key number (0-8)

"type" is
- L (local only)
- N (normal keyboard operation)
- T (transmit only)

"label is" any series of up to 8 characters in the base character set

"string" is any series of up to 80 characters

The soft key assignments are displayed in place of the normal screen display. Data in the display workspaces is not lost. (Note — if display memory is full, assigning characters to soft keys may cause some display data to be lost.) When the key assignment is completed and the terminal returned to normal operation, the old display is returned to the screen. Normal operation is restored using the following escape sequence:

Remove soft key menu: ESC k

Defining Soft Keys

The key assignment operation displays the current key assignments in format mode. The attribute, label, and string fields are unprotected allowing the operator to enter new values. In addition, the values are tested during input to make sure that only valid parameter values are used. When the terminal is initialized the soft keys are assigned default values as follows:

- [RETURN] = CR (normal)
- [f1] = ESC p (transmit only)
- [f2] = ESC q (transmit only)
- [f3] = ESC r (transmit only)
- [f4] = ESC s (transmit only)
- [f5] = ESC t (transmit only)
- [f6] = ESC u (transmit only)
- [f7] = ESC v (transmit only)
- [f8] = ESC w (transmit only)

NOTE: If the memory is full, adding characters to the soft key string may delete lines in normal display memory.

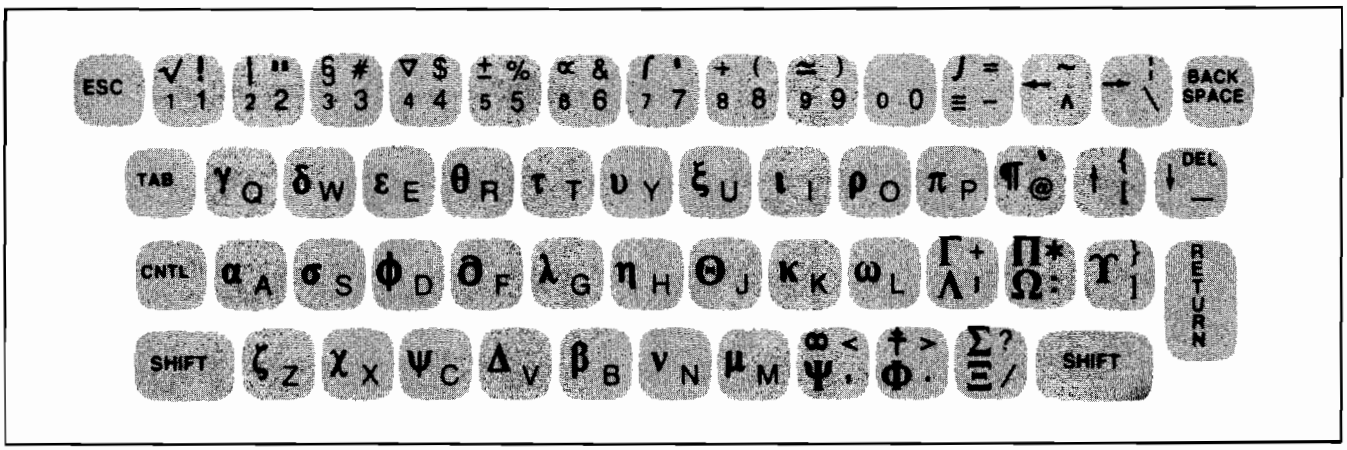


Figure 3-4. Math Set Elements

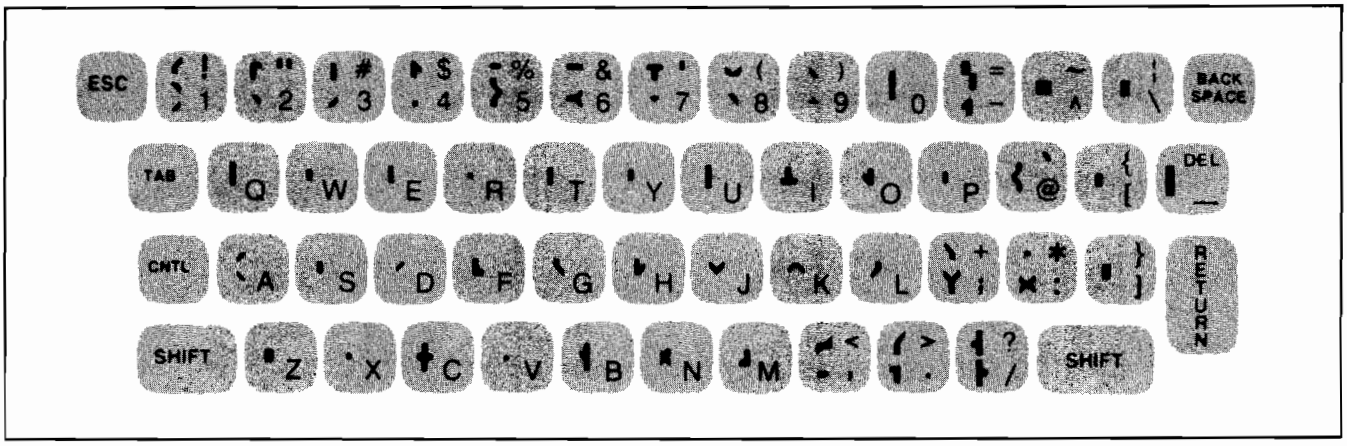


Figure 3-5. Large Character Set Elements

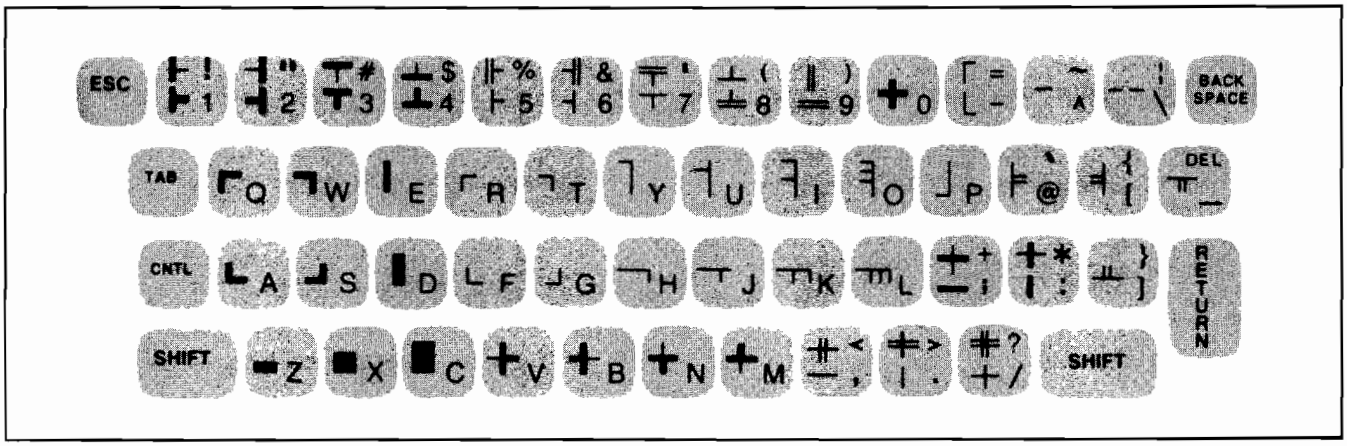


Figure 3-6. Line Drawing Set

Soft Key Applications

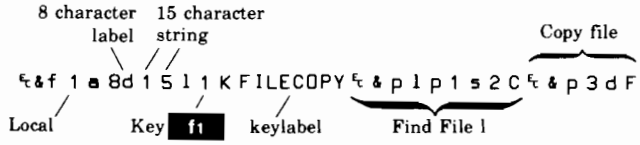
There are many applications of the soft keys. One application is the creation of "menu" operations. For example, the keys can be loaded with the escape sequences to find and read various files on a CTU emulator file set.

Example: Program the soft keys to find a file on the simulated left tape drive and copy the file to the display. This can be done using the following technique:

```
ESC&pnp1s2CESC&p3dF
```

where: n = the number of the file to be read

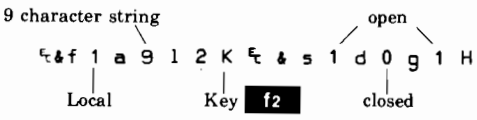
To program the f1 key to find and display file number 1 and label the key "FILECOPY":



The same procedure could be used for the remaining function keys.

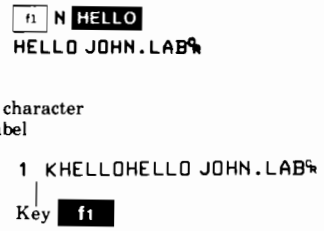
Another application would be to change the control settings on the Keyboard Interface to configure the terminal for use with different computer systems.

Example: To program the f2 key to change the settings of Switches D, G, and H to open, closed, and open, the following sequence could be used (no label):

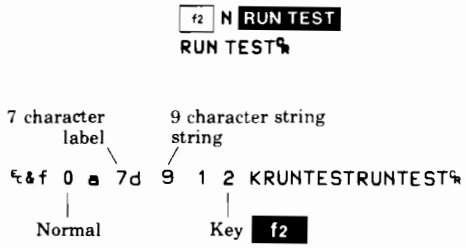


The soft keys can be used to hold log-on, program control, and log-off messages. This makes it easier for the terminal operator to use unfamiliar computer systems.

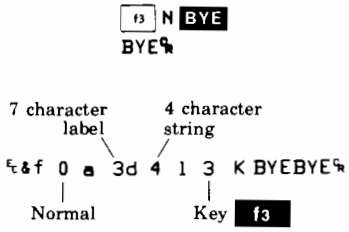
Example: A sample key assignment to log-on to the Hewlett-Packard 3000 Computer System might be as follows:



A second key could be used to call and execute a library program:



A third key could be used to log off of the system:



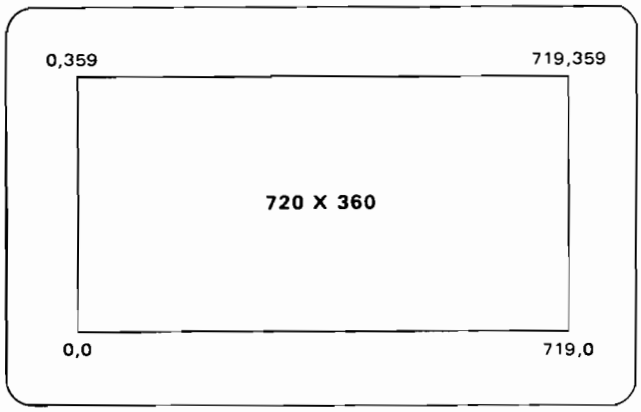
GRAPHICS CONTROL

SECTION 4

INTRODUCTION

This section contains a description of the terminals graphic capabilities and how to use them. It is divided into two main parts: control from the keyboard and control from a program. The graphics display can also be controlled using AGL (Another Graphics Language) which is described in the *BASIC/47 Manual*, part no. 02647-90038. Use of Multiplot, another graphics feature, is described in the *Autoplot/47 Manual*, part no. 02647-90042.

You can display graphics data by addressing points in a 720 by 360 array.



The graphics and alphanumeric data are displayed in the same area on the screen but are stored in separate RAM memories. This allows you to read or modify graphics and alphanumeric data separately.

NOTE: Display enhancements (blinking, inverse video, and half-bright) will affect the display of graphics data. The display enhancements affect alphanumeric character positions on the screen. Since the graphics data is displayed using the same screen dots as the alphanumeric data, it is also affected by the enhancements. The data in the graphics display memory is unchanged.

KEYBOARD CONTROL

All of the graphics function commands can be entered from the terminal keyboard by the operator. Most of the functions are available through a set of graphics control keys located to the right of the normal ASCII character set (see figure 4-1). Table 4-1 contains a list of the keys and a description of their functions. These keys can be used in both local and remote operation. This allows a combination of operator and program control of graphics functions to be used to make maximum use of the terminal's capabilities.

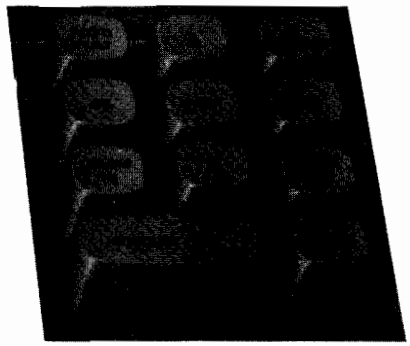


Figure 4-1. Graphics Control Keys

The keys used for graphics control have dual functions; as either graphics control keys or as numeric keys. The selection is made using the softkeys as follows:

[COMMAND], [next], [next], [next], [KEYPAD], [GRAPHICS], or [NUMERIC], [RETURN].

Note that the keys must be selected for graphics control to access the terminal's graphics capabilities.

Each key controls two functions. Pressing the key alone executes the command labeled on top of the key. Pressing both the function and shift keys executes the command labeled on the front of the key. The functions labeled on the front of the keys permanently alter the display (i.e., [CLEAR]). These functions are selected by pressing the [SHIFT] key and the function key. This makes a mistake less likely. The only graphics keys that repeat are those controlling zoom and the graphics cursor.

Table 4-1. Graphics Control Keys

KEY	DESCRIPTION	KEY	DESCRIPTION
AUTO PLOT	Turns autoplot on. If plot from display has been selected, or a point count specified, autoplot will turn itself off. Otherwise, the STOP key must be used to terminate autoplot.	A DSP	Toggles the alphanumeric display.
		CLEAR	Erases the graphics image memory.
G CURSOR	Toggles the graphics cursor on and off.	DRAW	Draws a vector from the current pen position to the graphics cursor. Only works if the cursor is on.
U D L R	Move the graphics cursor. More than one can be pressed for diagonal motion.	MOVE	Moves the pen to the graphics cursor without drawing a vector. The graphics cursor must be on.
CURSOR FAST	Speeds up the graphics cursor if pressed in conjunction with the cursor keys. The rate returns to normal when released.	TEXT	Selects the graphics image memory as the destination for all text. Characters entering from the keyboard, datacomm, or disc are drawn as vectors in the graphics memory, using the current size and angle as specified by the T SZE and T ANG keys. If the terminal is not in Compatibility Mode the drawing mode is set to jam pattern to allow for backspacing and retyping of characters. If the terminal is in Compatibility Mode the drawing mode is unchanged. (Both Drawing Mode and Compatibility Mode are discussed later in this section.)
ZOOM	Toggles zoom mode. When zoom is turned on, the area about the graphics cursor is magnified by the amount se by the ZOOM IN / ZOOM OUT keys. Moving the cursor changes the zoomed area.		
ZOOM IN	Increments the zoom magnification.		
ZOOM OUT	Decrements the zoom magnification.		The graphics cursor indicates the position of the next character. Moving the graphics cursor resets the start of line point. Carriage return, line feed, and backspace work as expected even on inverted text. The STOP key terminals this mode.
STOP	Turns multiplot and graphics text mode off.		
SHIFT	Key held down selects the alternate key functions.		
AXES	Draws the axes, tic marks, and labels specified in the autoplot menu.	T SZE	Increases the character size from 1 to 8X. The smallest character is a 5 by 7 matrix in a 7 by 9 cell. Increasing the size makes the dots bigger; the character is still drawn as a 5 by 7 matrix.
AUTOPLOT MENU	Toggles the menu for autoplot parameters.		
RB LN	Toggles the rubber band line connecting the current pen position and the graphics cursor.	T ANG	Sets the character orientation (multiples of 90 degrees) and turns slant on or off.
G OSP	Toggles the graphics display, to inhibit the graphics image without erasing.		

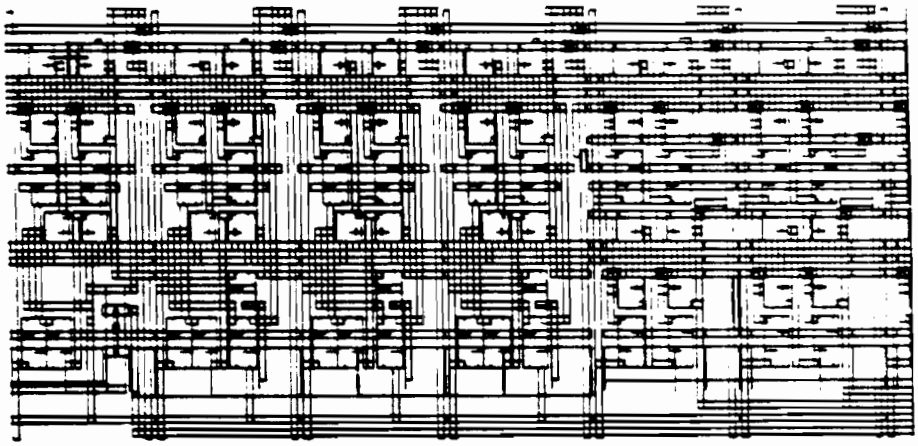
Display Control

CLEAR/SET GRAPHICS IMAGE MEMORY. The graphics image memory can be completely cleared (screen all black) or set (screen all white) independent of the alphanumeric display memory. SHIFT CLEAR will clear the display, and typing ↑c * d B will set the display.

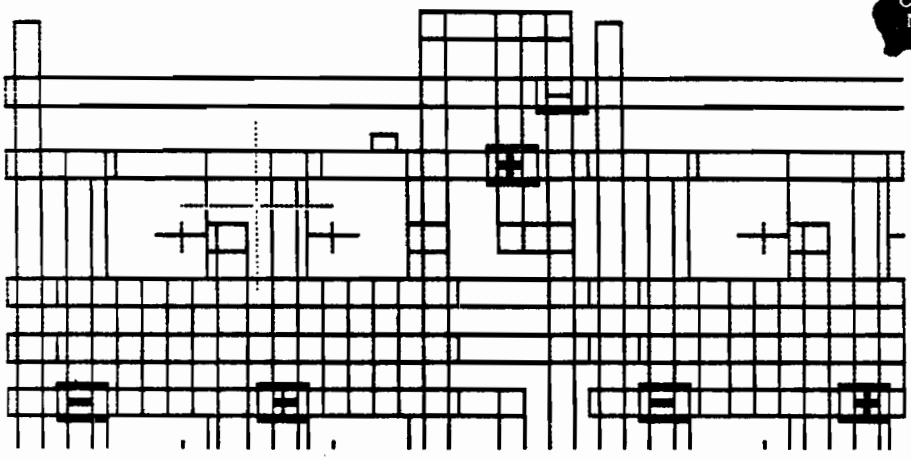
GRAPHICS VIDEO AND ALPHANUMERIC VIDEO ON/OFF. The graphics memory display can be turned on and off by SHIFT G DSP. The alphanumeric memory display can be turned on and off by SHIFT A DSP. Information stored in either is not lost when the other video is turned off. These keys allow either memory to be displayed; both may be on or off simultaneously.

ZOOM ON/OFF. Any portion of the graphics display can be expanded from 1 to 16 times. ZOOM IN incrementally expands the display at the cursor position, while ZOOM OUT incrementally reduces the display at the cursor position. The ZOOM key will toggle between the normal 1:1 display and the expanded display size set by the ZOOM IN, ZOOM OUT keys. (Illustrations below show an IC mask at 1:1, the upper left corner magnified four times, and the same area magnified 11 times.

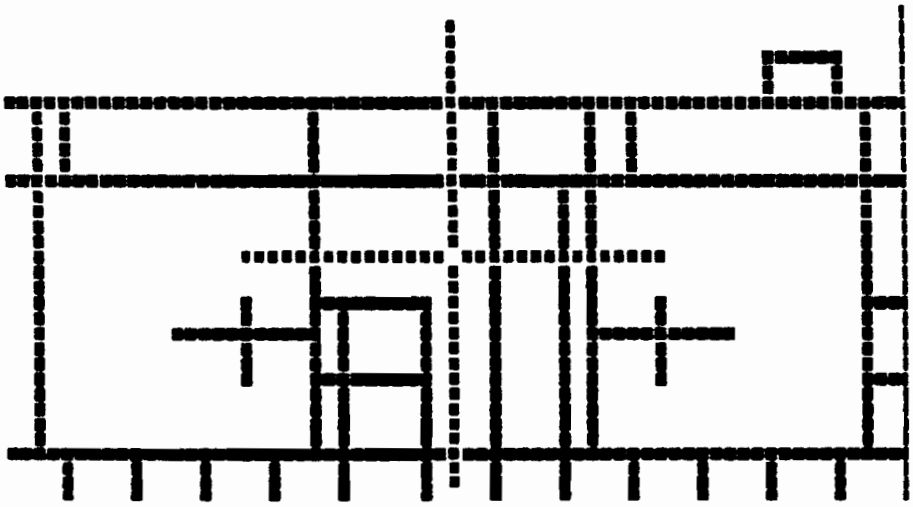
PANNING. With the display expanded by the zoom keys, the graphics cursor control keys allow you to view any portion of the graphics display. When the cursor reaches the edge of the screen, a portion of the graphics display is moved into the screen viewing area. This allows you to "pan" throughout the entire graphics memory at any zoom size.



IC Mask At 1:1



Upper Left Corner Magnified 4 Times



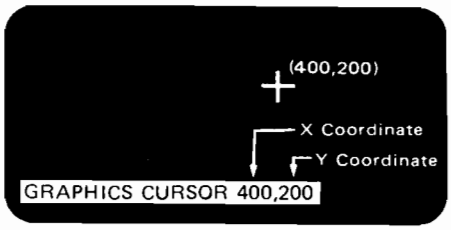
Upper Left Corner Magnified 11 Times

Cursor Control

ALPHANUMERIC CURSOR. The alphanumeric cursor is turned on and off when the alphanumeric memory is turned on and off (SHIFT A DSP). However, typing ESC * d R will turn off the cursor, and ESC * d Q will turn it on. This feature is useful for making hard copies of the display with the cursor off. The cursor is also turned on whenever the alphanumeric video is turned on.

GRAPHICS CURSOR. The graphics cursor may be turned on and off by G CURSOR . The cursor will not be displayed, however, if the graphics video is turned off. The cursor arrow keys on the keypad move the cursor to any position on the display. The CURSOR FAST key may be held down while pressing the cursor arrow keys to increase the cursor move speed. Any two cursor keys at right angles to each other may be held down simultaneously to move the cursor diagonally.

GRAPHICS CURSOR POSITION. You can find out the exact position of the graphics cursor at any time by CNTL G CURSOR . This will cause the cursor coordinates to be displayed in the lower left corner of the screen.



The display of cursor coordinates will remain when the graphics cursor keys (LEFT , UP , RIGHT , DOWN) are used to move the cursor. The displayed coordinate values will change as the cursor is moved. This allows you to easily position the cursor to any predetermined point.

Once the cursor is correctly positioned simply press the G CURSOR or RETURN key to return the normal display. You must return to the normal display to perform any of the graphics drawing functions (MOVE , DRAW , etc.).

RUBBER BAND LINE. Rubber band line is toggled by SHIFT RB LN . The *rubber band line* feature allows you to "preview" lines before you draw them. The rubber band line is displayed from the current "pen" position to the graphics cursor. As you move the cursor, the line will stretch and contract like a rubber band from the current "pen" position.

"Pen" Control

To draw a vector from one point on the display to another, the "pen" must be moved to the beginning of the vector (SHIFT MOVE or SHIFT DRAW). The graphics cursor is then moved to the desired end point and the line is drawn (SHIFT DRAW).

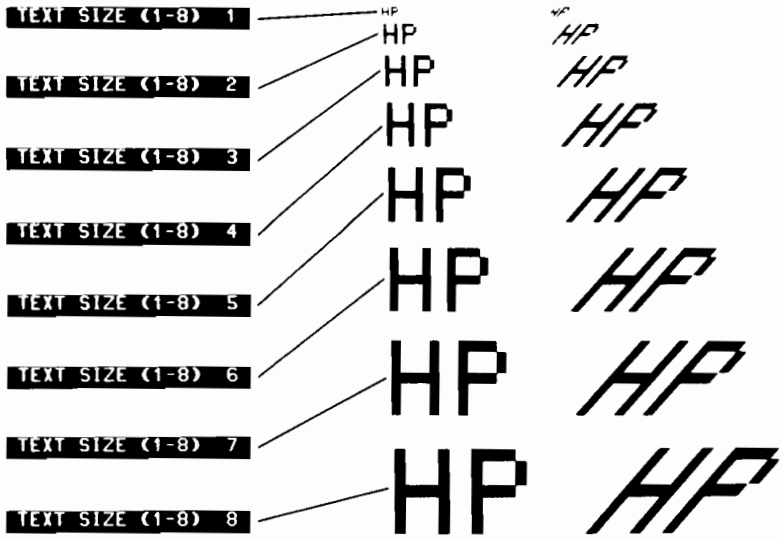
Graphics Text Mode

The 96 displayable characters on the keycaps in the Character Set Group can also be entered into the graphics memory. The characters can be displayed either *upright* or *slanted* (like italics) in eight different sizes. Also, the text lines can be displayed at four different angles (90 degree increments).

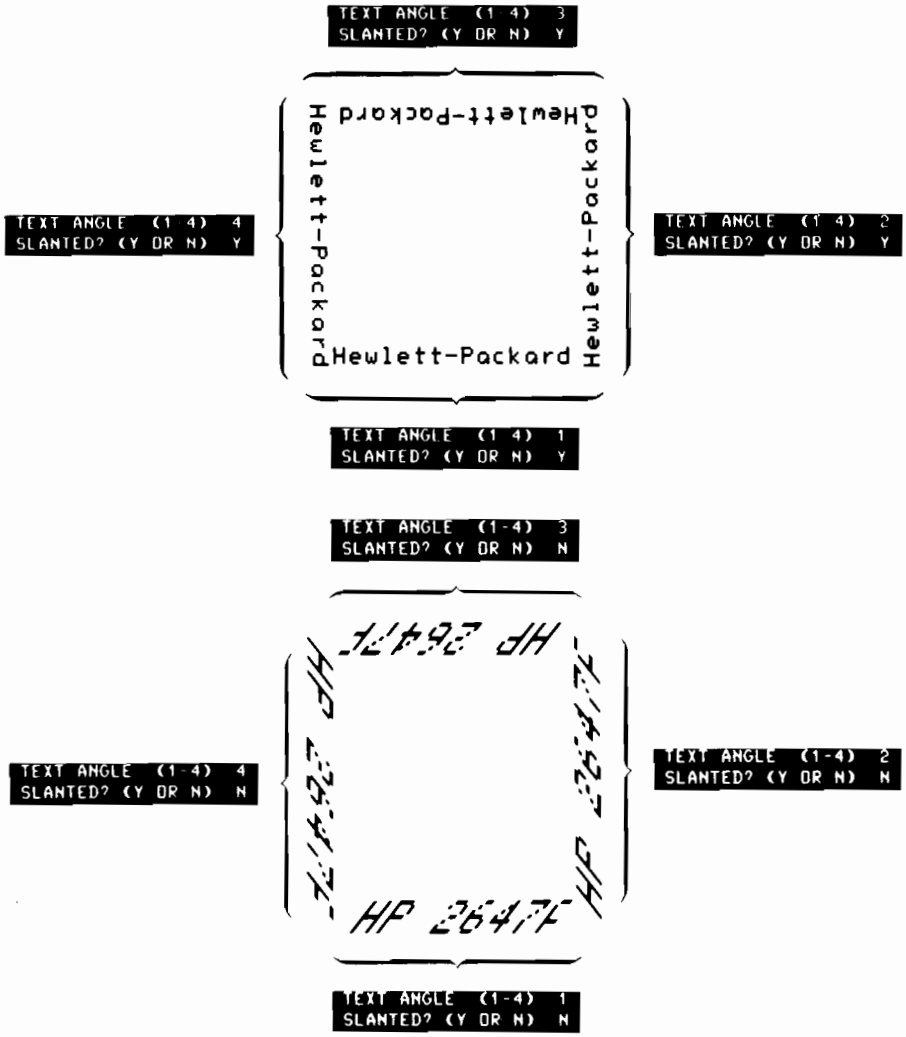
TURNING ON GRAPHICS TEXT MODE. Press SHIFT TEXT to turn on graphics text mode. Characters typed at the keyboard in local mode or received from the computer in remote mode will be displayed starting at the current cursor position. The current graphics cursor position will also define the left margin.

The keys used in graphics test mode are shown highlighted at the left. In addition to the normal 96-character keys, the RETURN , BACK SPACE , and SHIFT keys can be used. The TAB key functions like the alphanumeric cursor right (RIGHT) key. Editing keys (INSERT LINE , DELETE LINE , INSERT CHAR , DELETE CHAR) do not function in graphics memory.

TEXT SCALE FACTOR. The size of the text characters is specified by typing SHIFT T SIZE . The message shown below will appear on the display, then you enter any number from "1" to "8" to specify the size of the characters. Pressing RETURN resumes the normal display.



TEXT ANGLE (90-DEGREE INCREMENTS) AND CHARACTER ANGLE (ITALICS). The text angle is specified by typing [SHIFT] [T ANG]. The message shown below will appear on the display, then you enter a number from "1" to "4" for the desired angle.



Normal (upright) and italic (slanted) characters are specified by typing "Y" (yes) or "N" (no) to the SLANTED? message.

Either parameter may be entered first, or only one of them entered when the message is displayed. Pressing the RETURN key resumes the normal display.

JUSTIFICATION (LEFT, RIGHT, CENTER). Text may be *left-justified* (normal, default), *center-justified,* or *right-justified* as shown at right.

NOTE: When using *center-justify* or *right-justify,* the line of text is not displayed until RETURN is pressed after typing the line.

Left-Justified:
```
Hewlett-Packard
19400 Homestead Road
Cupertino, California 95014
```

Center-Justified:
```
        Hewlett-Packard
      19400 Homestead Road
   Cupertino, California 95014
```

Right-Justified:
```
                Hewlett-Packard
              19400 Homestead Road
           Cupertino, California 95014
```

Also, the text/cursor relationship may be *bottom* (normal, default), *center,* or *top.* You must type escape codes from the keyboard to access these features. The list of escape codes are given at right and are shown in the illustrations.

Left Justify, cursor left-bottom of character	&c•m0Q	Hewlett-Packard
Left Justify, cursor bottom	&c•m1Q	Hewlett-Packard
Left Justify, cursor center	&c•m2Q	Hewlett-Packard
Left Justify, cursor top	&c•m3Q	Hewlett-Packard
Center Justify, cursor bottom	&c•m4Q	Hewlett-Packard
Center Justify, cursor center	&c•m5Q	Hewlett-Packard
Center Justify, cursor top	&c•m6Q	Hewlett-Packard
Right Justify, cursor bottom	&c•m7Q	Hewlett-Packard
Right Justify, cursor center	&c•m8Q	Hewlett-Packard
Right Justify, cursor top	&c•m9Q	Hewlett-Packard

CURSOR CONTROL. The *graphics cursor keys* control where the text will begin and provide a reference by the RETURN, BACK SPACE, and alphanumeric cursor keys. If the graphics cursor is moved by the *graphics cursor control keys* after some text has been entered onto the display, a new reference point will be established.

The *alphanumeric cursor keys* move the graphics cursor in while character spaces so that the cursor will be positioned correctly for editing the text already on the display, etc.

Therefore, use the graphics cursor keys to provide a starting reference point, then use the alphanumeric cursor keys, RETURN, AUTO LF, BACK SPACE and space bar thereafter to ensure correct positioning.

TERMINATING GRAPHICS TEXT MODE. When you have completed the graphics text, you may want to use the Alphanumeric Mode. Pressing the STOP key terminates the graphics text mode and allows alphanumeric text to be entered. *You may have to press* SHIFT A DSP *to return the alphanumeric cursor to the display.*

Transferring Graphics Data

The contents of graphics memory can be transferred to a disc file and/or to a compatible printer using either the Copy or Transfer command. Also, graphics data on a disc file can be restored to graphics memory using the same commands. Graphics data exists in 8-bit binary form.

As an example, the following keys can be pressed to transfer all the contents of graphics memory to a disc file named "DRAIN".

COMMAND , next , next , TRANSFER , ALL , from GRAPHICS , TO:file , drain, RETURN .

The command Copy could have been used in place of the command Transfer.

To return the data from file "Drain" to graphics memory from the keyboard, the following keys could be used:

COMMAND , COPY , ALL , from drain, to , GRAPHICS , RETURN .

In this case, the Transfer command could have been used in place of the Copy command.

Erasing the Display

Pressing SHIFT CLEAR will erase all graphics from the display.

NOTE: It is possible to enter graphics text into the graphics memory without seeing it on the display. Pressing SHIFT G DSP toggles the graphics display, and if the terminal is in the non-display mode, you may enter text without seeing it. Pressing the SHIFT G DSP keys again will display the text.

Also, the G CURSOR key toggles the graphics cursor display. If the cursor is not displayed, you may still enter text onto the display. However, the cursor will not be shown, but the text will be. The graphics cursor keys will not move the graphics cursor when it is not displayed in graphics text mode. The alphanumeric cursor keys, BACK SPACE, RETURN, and AUTO LF will move the graphics cursor whether on or off in graphics text mode.

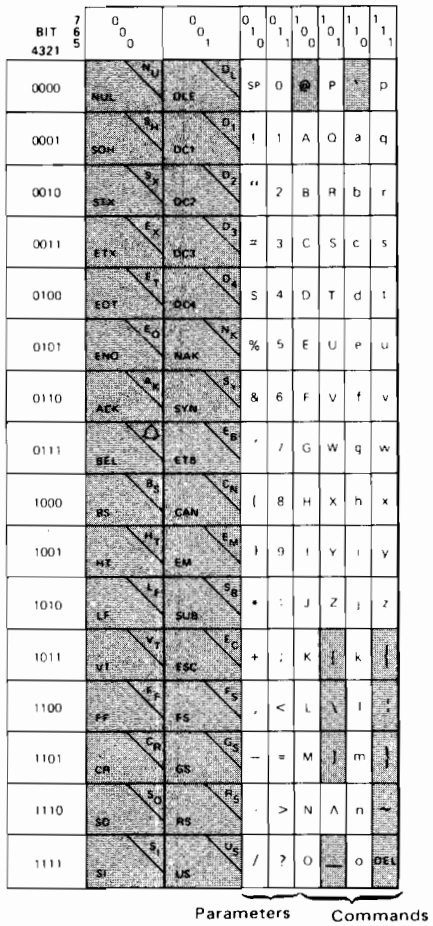
PROGRAM CONTROL

Graphics functions can be controlled by parameterized escape sequences. All graphics escape sequences begin with ESC *. The third character, always lower case, selects the type of graphics sequence. Table 4-2 lists the types of graphics sequences. For example, ESC * p specifies a plotting sequence.

Table 4-2. Summary of Graphics Sequence Types

ESCAPE SEQUENCE	DESCRIPTION
ESC * p	Vector Plotting
ESC * m	Vector Drawing Mode
ESC * d	Display Control
ESC * l	Labeling
ESC * t	Compatibility Mode
ESC * s	Graphics Status
ESC * r	Raster Dump (device control)
ESC * b	Raster Dump (data transfer)

Subsequent characters in the control sequence are read as either parameters or commands, depending on the location of the character in the ASCII table.



BIT 7 6 5 4 3 2 1

0 0	Control Code
0 1	Parameter
1 0	Command and Terminate Sequence
1 1	Command and Continue Sequence

Control Codes

Control codes are generally ignored, with the exception of ESC. If an ESC character is detected and the previous graphics control sequence has not been properly terminated with a "Z" or some other valid capital character, the ESC will cause the execution of the previous sequence to be terminated. The new escape sequence will then be executed.

Commands

Graphics commands come from columns 4-7 of the ASCII table, the upper and lower case letters (A-Z and a-z). Both upper and lower case commands execute the same function. Upper case letters terminate the sequence and cause it to be executed. You can use more than one command in a sequence.

Graphics sequences can be any length. (The terminal ignores DEL and NUL characters in the middle of graphics sequences.) For example, to plot a figure containing 100 points the escape sequence could appear as follows:

ESC * p a <x1,y1> . . . <x100,y100>Z

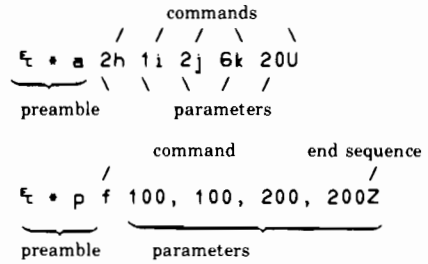
This could cause problems if an error occurs and the system tries to report it in the middle of a long sequence. Since most systems use upper case characters for messages, the first character of the message would end any graphics sequence that might be in progress. Letters that have not been assigned a function for a particular graphics sequence are treated as NOPs and if they are lower case, are ignored. If upper case, they will end the sequence. The letter z has ben defined as a NOP in all sequences so that a capital Z can always be used to end a graphics escape sequence.

Parameters

Parameters come from columns 2 and 3 of the ASCII table (SPACE through ?). Most parameters are simply the ASCII numeric characters used to represent data co-ordinates or to select one of several settings. Binary formatted data is generated by appending the bits 0 1 to five bits of binary data. Note that in binary formats, spaces are treated as data and are not ignored or used as delimiters. Both ASCII and binary data formats are described later in this section.

Parameters precede their associated commands (postfix notation). The most frequently used parameters are vector data. Refer to the discussion of Vectors for additional information on parameters used to define vector operations.

Examples:



The programmable graphics functions are organized into five major groups.

- Graphics Display Control
- Plotting
- Raster Dump
- Graphics Text
- Compatibility Mode

The remainder of this section contains descriptions of each of these functional groups.

Graphics Display Control

Graphics display control is made up of the functions used to control the graphics cursor, the portion of the graphics memory that is currently being displayed, or the state of the graphics memory. These functions are as follows:

- Graphics Cursor Control
- Zoom
- Graphics Memory Control

Table 4-3 lists the escape sequences for each of the graphics display control functions.

Table 4-3. Graphics Display Control Functions

FUNCTION	CODE	DESCRIPTION
Graphics Cursor Control		
Cursor On	Ec*dK	Turn on the graphics cursor.
Cursor Off	Ec*dI	Turn off the graphics cursor.
Move Absolute	Ec*d<x,y>o	Position the graphics cursor.
Move Relative	Ec*d<x,y>p	Position the graphics cursor.
Zoom		
Zoom On	Ec*dg	Turn on the zoom function.
Zoom Off	Ec*dh	Turn off the zoom function.
Zoom Size	Ec*d<size>i	Set the zoom size.
Zoom Position	Ec*d<x,y>j	Set the zoom position.
Graphics Memory Control		
Clear Memory	Ec*da	Turn off all dots in graphics memory.
Set Memory	Ec*db	Turn on all dots in graphics memory.
Display On	Ec*dc	Enable the graphics display.
Display Off	Ec*dd	Inhibit the graphics display.

GRAPHICS CURSOR CONTROL. A separate graphics cursor is available for use in locating points in the graphics display. The graphics cursor is used by the terminal operator to input position data or to interact with a graphics application program.

Graphics Cursor On/Off. The graphics cursor is initially off (power on or full reset). Turning the cursor on or off does not affect the data in graphics memory. (The graphics cursor is also turned on when graphics text operation is enabled.)

Graphics Cursor On: Ec*dK

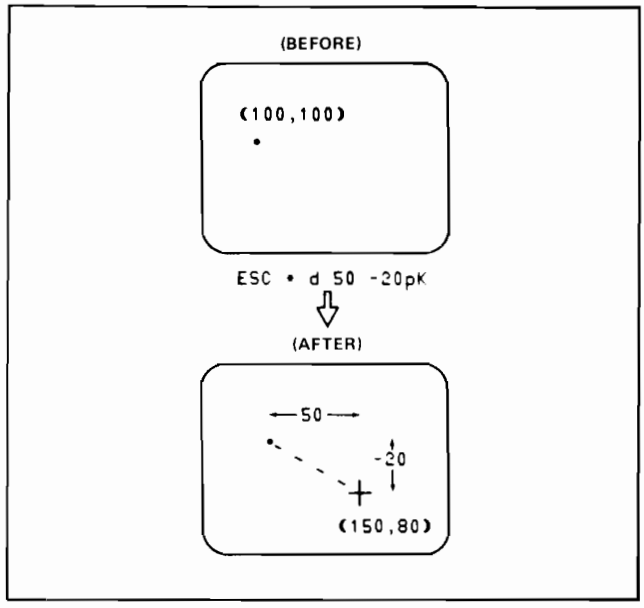
Graphics Cursor Off: Ec*dL

Graphics Cursor Positioning: The graphics cursor is initially at position (0,0) after power on or a full reset. The cursor can be positioned (even if it is not turned on) using either absolute or relative coordinates. In the following sequences X and Y give the new cursor position. Refer to Vectors for a discussion of absolute and relative coordinates.

Position Graphics Cursor Absolute: Ec*d<X,Y>o

Position Graphics Cursor Relative: Ec*d<X,Y>p

Example: The cursor is currently at position 100,100 and off. Move it 50 units to the right and 20 units down from its current position and turn it on.



ZOOM. You can display a portion of the graphics memory data at increased size. The zoom size (magnification) settings are from 1 to 16 times. The data in the graphics memory is not lost or changed. As much of the graphics memory as will fit on the screen at the new size is displayed. The effect is similar to "windowing". The portion of the graphics memory to be centered on the screen can be changed allowing you to "pan" through the entire graphics memory at the magnified setting.

Zoom Size. The zoom size is initially set to 1 after power on or a full reset. The size can be set to one of sixteen sizes (1-16). The magnified data is not displayed until the zoom function is turned on.

Set Zoom Size: `ESC*d<size>I`

where: `<size>` is 1-16

Zoom Position. The zoom position is initilly the graphics cursor position. It can be set to any position in graphics memory using ASCII absolute coordinates. The selected data is not displayed until the zoom function is turned on.

Set Zoom Position: `ESC*d<X,Y>J`

If multiple images are being displayed by changing the zoom position to display only a portion of the screen at a time, it is possible to insert a delay of approximately 16 ms between the images (frames) by inserting the "zoom on" command between positioning commands. (Refer to Inserting Delays In Graphics Operations.)

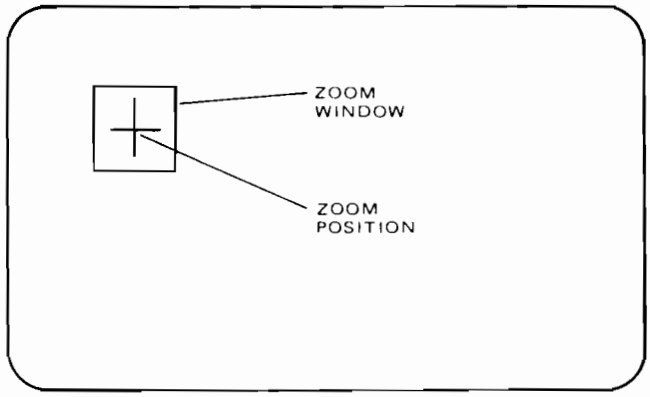
Example: Step the zoom position between (50,50), (50,250), (250,250), and (250,50) at 16 ms intervals.

```
ESC * d 50,50J g 50,250J g 250,250J g 250,50J g 50,50J
```

Zoom On/Off. Once a zoom size and position are selected the data is displayed by turning on the zoom function. If the cursor is outside of the zoom window, turning on the zoom function causes the graphics cursor to be moved to the zoom position.

Data displayed on the screen while the zoom function is on can be modified using the graphics memory set and clear commands. Most of the data in the graphics memory that is not being displayed is unaffected by the set and clear memory commands (see the note in the following example) but can be modified using the plotting commands described later in this section.

Moving the graphics cursor while zoom is turned on will cause the display to "pan" across the data in the graphics memory. If the graphics cursor is moved beyond the edge of the "zoom window" the window will move with the cursor. This causes the zoom position to change accordingly.



NOTE: Up to 16 points to the left and right of data displayed in the zoom window may be affected by the set or clear commands.

Example: Set the zoom size to 8, center the zoom window at 100,200 and turn on the zoom function.

```
ESC*d 8I 100 200Jg
```

GRAPHICS MEMORY CONTROL. The graphics display can be turned on or off or the entire memory can be set to all ones (dots on) or all zeros (dots off).

Graphics Display On/Off. The graphics display and graphics cursor can be turned on or off. The data in the graphics memory is unaffected.

Graphics Display On: `ESC*dc`

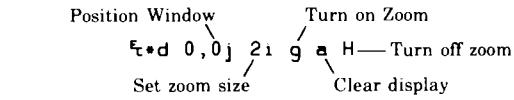
Graphics Display Off: `ESC*dd`

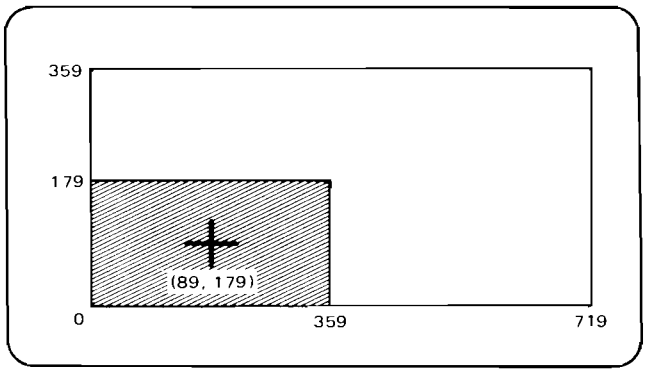
Graphics Display Set/Clear. The graphics data currently displayed on the screen can be set to all ones or cleared to all zeros. When used together with the zoom function this function can be used to clear or set blocks of the graphics memory.

Clear Graphics Memory: `ESC*da`

Set Graphics Memory: `ESC*db`

Example: Clear the lower left portion of the graphics display.





Inserting Delays in Graphics Operations

Certain graphics operations are executed only at the end of a frame (drawing of the display). These operations take approximately 16 ms and can be used as delays to slow down or synchronize changes in the display. Refer to Set Zoom Position for an example.

OPERATION	CODE
Graphics Display On	Esc*dc
Graphics Display Off	Esc*dd
Alphanumeric Display On	Esc*de
Alphanumeric Display Off	Esc*df
Zoom On	Esc*dg
Zoom Off	Esc*dh
Set Zoom Size	Esc*di
Set Zoom Position	Esc*dj

Graphics Drawing Mode Parameters

There are several drawing parameters that can be set to allow a wide variety of drawing capabilities. These parameters select whether data will be stored in the graphics memory as 1's or 0's, define line or area patterns to be used when drawing vectors, position the relocatable origin, and define graphics text settings.

Graphics drawing control sequences begin with Esc * m followed by one or more of the drawing parameters. Table 4-4 lists the mode control commands.

DRAWING MODES. Vectors can be drawn by setting, clearing, or complementing the data in the graphic memory. Normally the memory is cleared and vectors are drawn by setting selected bits to make white lines on a dark screen. If instead you want black vectors on a white screen, you can begin by setting memory (refer to the Set Memory command), select a clear or complement line type and draw dark vectors (refer to the example that follows). Figure 4-2 illustrates the various drawing modes.

Table 4-4. Graphics Mode Commands

Esc * m <parameters>	
PARAMETERS	**DESCRIPTION**
a	select drawing mode
b	select line type
c	define line pattern
d	define area pattern
e	area fill, absolute
f	area fill, relocatable
j	set relocatable origin
k	set relocatable origin to pen position
l	set relocatable origin to cursor position
m	set graphics text size
n	set graphics text direction
o	turn on character slant
p	turn off character slant
q	set text origin
r	set graphics defaults
z	NOP

Set Drawing Mode: Esc*m <parameter> a

where: <parameter> is

0 Graphics memory not changed.
1 Clear (turn off graphics bits).
2 Set (turn on graphics bits).
3 Complement (toggle the graphics bits).
4 Jam (turn bits on or off according to the data).

Clear Mode. Clear mode causes selected display bits to be turned off. The "selected bits" are those that are "on" in the line pattern. If a solid line type (the default) has been selected, all of the bits in a vector will be selected. In clear mode this means that all of the dots making up a vector will be turned off. This allows you to draw dark vectors on a white background. Only those bits that are on in the pattern are cleared. Bits that are off in the pattern do not affect the display.

Set Mode. Set mode is similar to clear mode except that the selected bits are turned on instead of off. Only the bits that are on in the line type are affected.

Complement Mode. Complement mode causes the selected display bits to change state (on to off, off to on). Again only those bits that are on in the line type or pattern are affected.

Jam Mode. Jam mode differs from the other modes in that both the bits that are on in the line type or pattern and the bits in the pattern that are off affect the display. Jam mode has the effect of overlaying the display with the pattern.

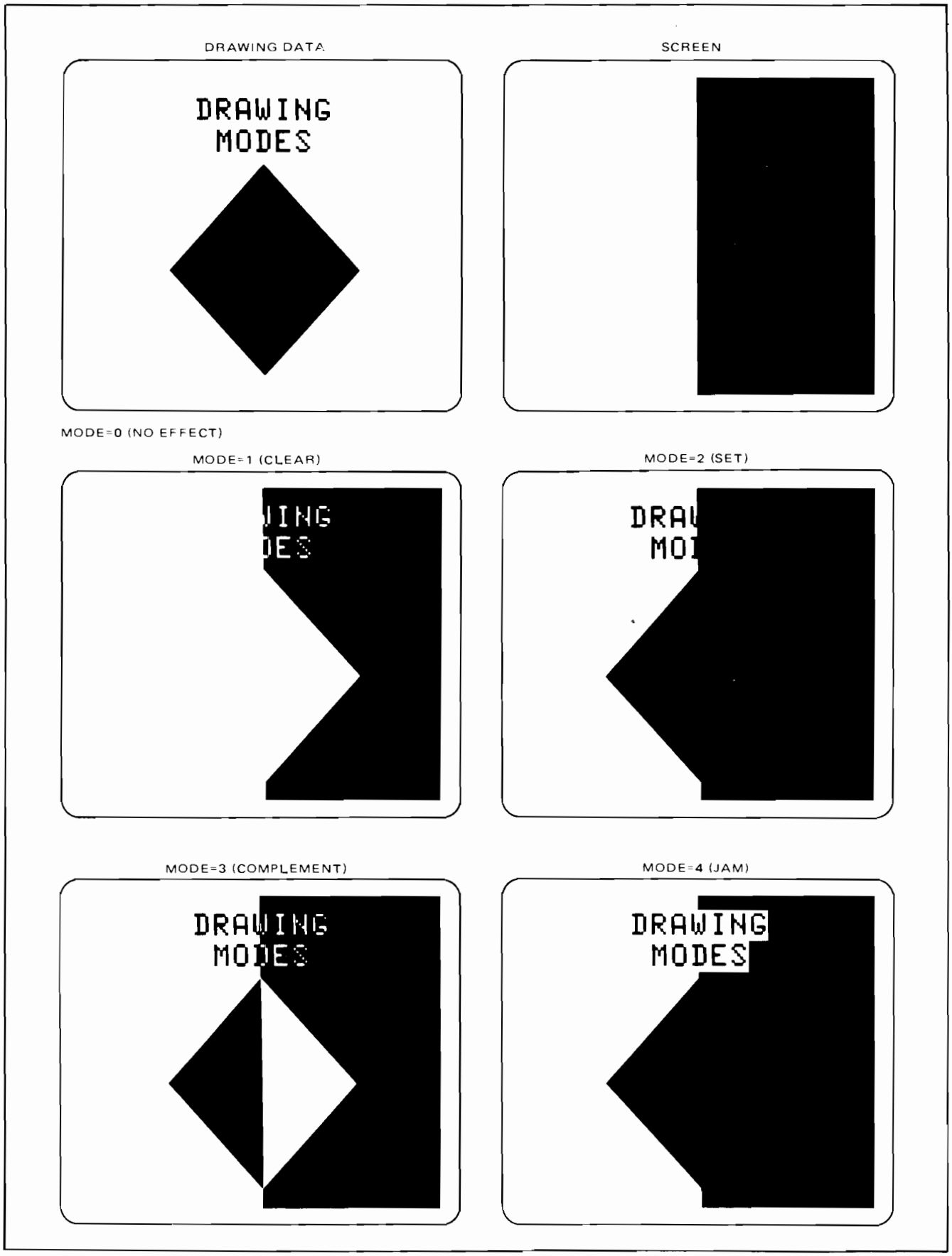


Figure 4-2. Examples of Drawing Modes

Selective Erase. A vector drawn in set mode can be selectively erased by redrawing it in clear mode. This will cause gaps to occur if the erased line is intersected by other lines. This problem can be overcome by initially drawing the line in complement mode and then redrawing it in complement mode to erase the line. This technique will preserve the original display. Complement mode is useful for drawing and erasing temporary figures.

Example: Select complement mode, draw a vector, and then erase the vector by redrawing.

```
^t*m3A      (select complement mode)
^t*pa f 100,300 300,300Z      (draw vector)
^t*pa f 100,300 300,300Z      (erase vector)
```

DRAWING PATTERNS. You can select the dot pattern used when drawing vectors or filling rectangular areas. Dotted and dashed lines can be drawn by selecting one of nine predefined line patterns or a user defined line or area pattern. This allows you to use different line patterns to distinguish between groups of plotted data or easily generate shading and cross hatching for use in engineering drawings, graphs or fabric patterns.

Line Type. One of eleven line types can be selected. Once a line type has been selected all drawing vectors are drawn using that line type. The patterns for the predefined line types are shown in figure 4-3. Refer to the Define Line Pattern command for additional information.

Select Line Type: `^t*m<line type>b`

where: `<line type>` is

1 Solid line (default)
2 User defined line pattern
3 User defined area pattern
4 Predefined pattern #1
5 Predefined pattern #2
6 Predefined pattern #3
7 Predefined pattern #4
8 Predefined pattern #5
9 Predefined pattern #6
10 Predefined pattern #7
11 Point plot

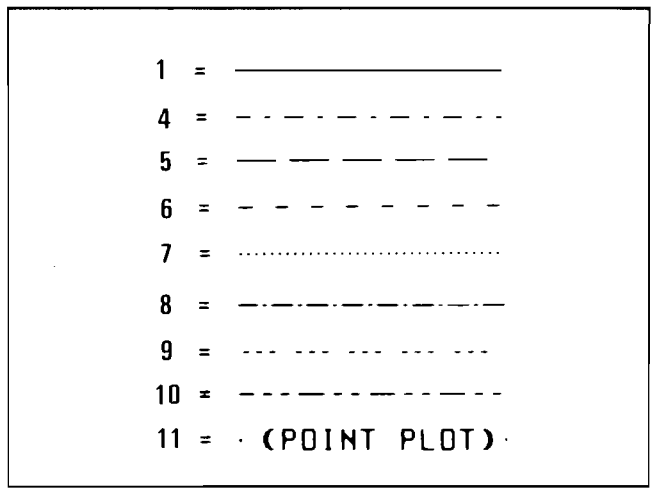
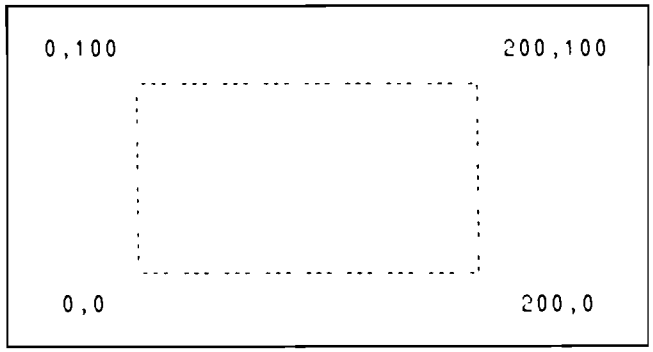


Figure 4-3. Predefined Line Type Patterns

Point plot causes a single point to be plotted at the coordinates specified by the data. This line type is useful for generating "scattergram" type graphs. If user defined area shading is selected (type = 3) the line patterns used are selected from the eight lines making up the area fill pattern (refer to Define Area Pattern). The display is divided into groups of eight rows and eight columns. Horizontal and vertical lines are drawn using the appropriate row or column from the area pattern. Diagonal lines are drawn using a solid vector.

Example: Select line type 9 and draw a figure using the new line type.

```
^t*m9B
^t*pa0,0 200,0 200,100 0,100 0,0Z
```



Example: Select the area pattern as the line type.

```
ESC * m 3 B
```

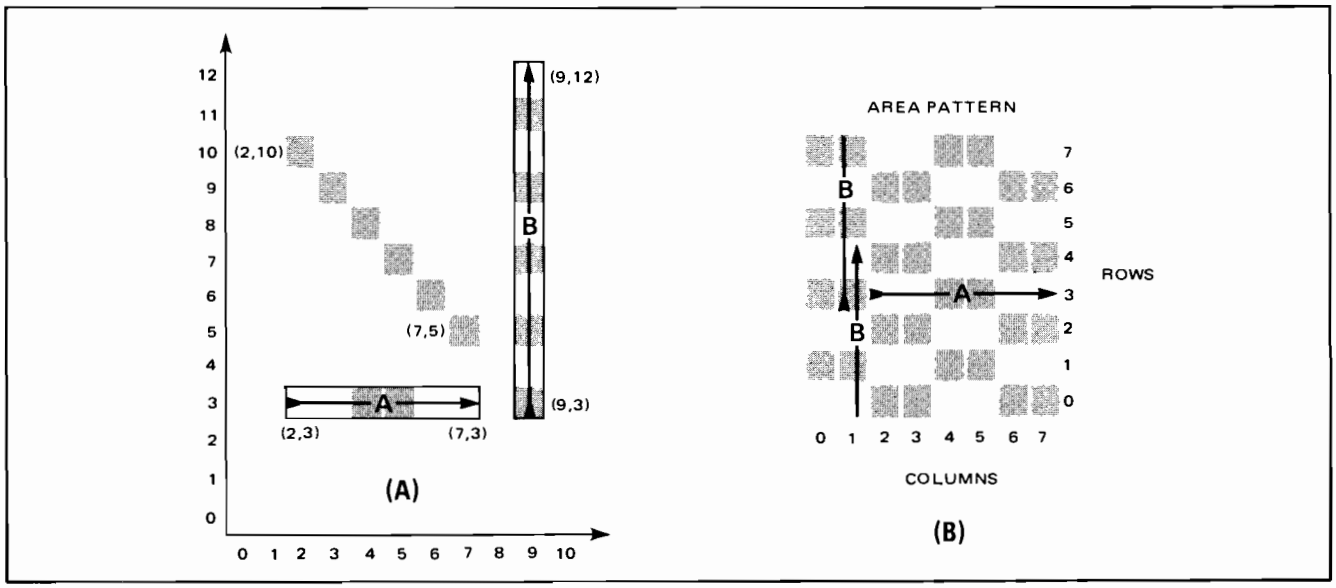


Figure 4-4. Using Area Patterns As Line Types

Drawing vectors (2,3)-(7,3), (9,3)-(9,12), and (3,7)-(7,3) using the area pattern shown in figure 4-4b would result in the drawing shown in figure 4-4a.

Adjacent horizontal or vertical lines using the user defined line type (type = 2) can be used to create patterns more complicated than those available in an 8×8 area pattern. User defined line and area patterns are described in the following paragraphs.

Define Line Pattern. The dot pattern used to draw vectors can be defined programmatically. Once a pattern is defined you must select the user defined line type (type = 2) using the Select Line Type command. Figure 4-5 gives examples of line patterns.

Define Line Pattern:

```
ESC * m <pattern> <scale> c
```

where: <pattern> is the decimal value 0 to 255) defining an 8-bit binary pattern. For example,

. . . . = 10101010 = 170.

<scale> is a scale factor (1 to 16) to be applied to the pattern. For example, with a scale factor of 3 the pattern defined above would be as follows:

... ...

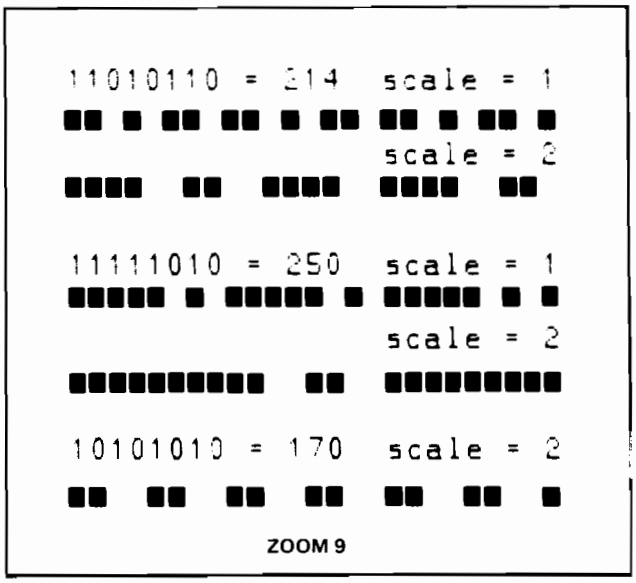
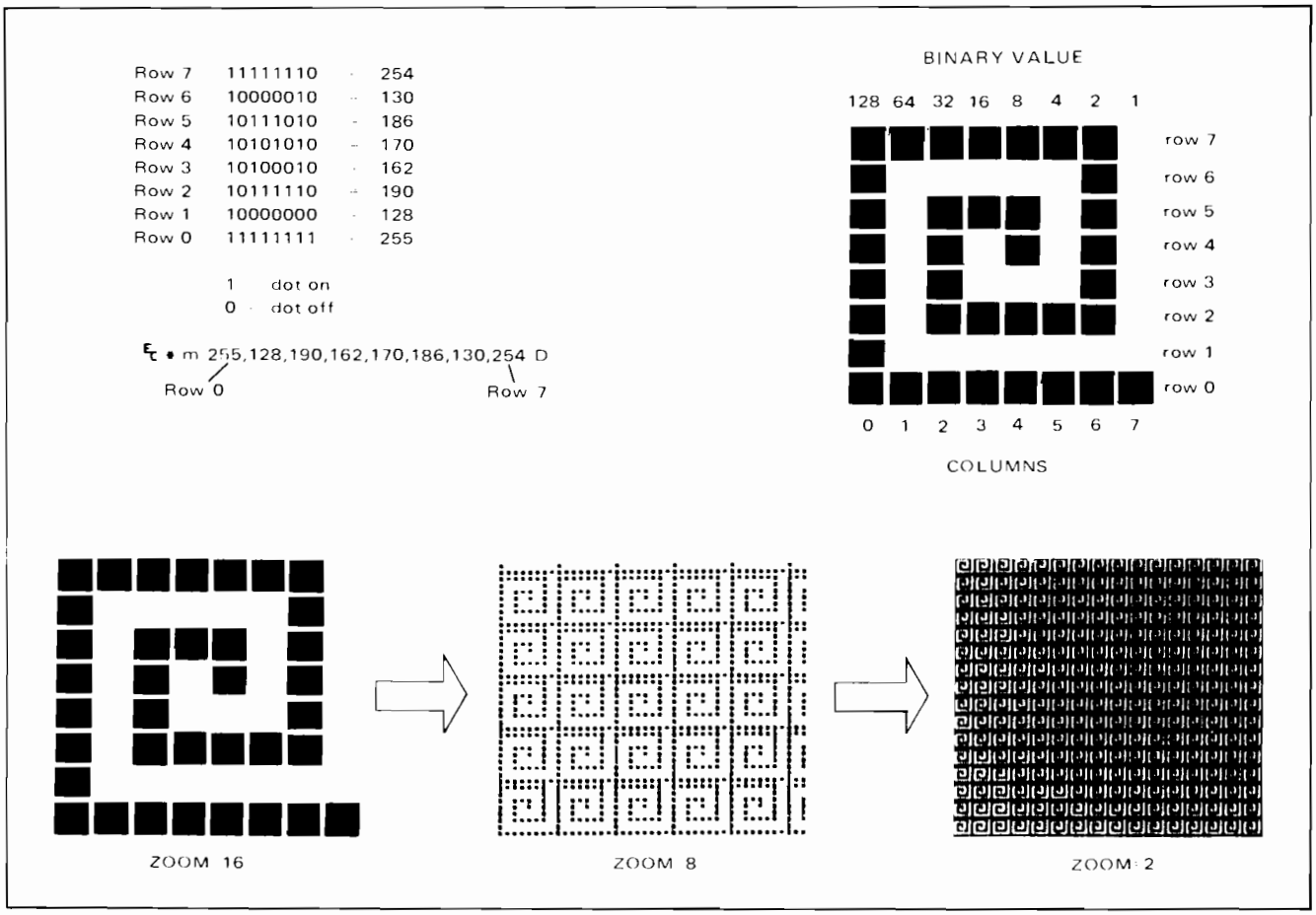


Figure 4-5. Examples of User Defined Line Patterns

Line patterns too complex to be obtained from an 8×8 area pattern can be generated by plotting a series of lines and varying the patterns used for successive lines. Complex patterns such as those used in weaving can be generated easily using this technique.

Example: Define a pattern to generate the following vector:

```
**********oo**oo**********oo**oo
pattern = 111110100 = 250
scale   = 2
ESC * m 250 2 C
```



Define Area Pattern. An 8×8 pattern can be defined for use in filling rectangular areas. The pattern can also be used to provide line patterns for horizontal or vertical lines when the area pattern is selected as a line type (type=3). (Refer to Define Line Type.) Irregular shapes can also be built up by selecting the area shading pattern and then using successive lines.

The area pattern is defined using 8 parameters, one for each of the rows in the pattern. Each parameter is a decimal number (0 to 255) representing an 8-bit binary pattern. Refer to Define Line Pattern for additional information. The display is divided up into 8×8 cells. Every point on the display is mapped to a corresponding bit in the pattern. Drawing horizontal or vertical lines causes the corresponding row or column of the pattern to be used as the line pattern. Diagonal vectors will always be drawn using a solid line. Figure 4-6 contains sample area fill patterns.

Define Area Pattern:

```
ESC * m <row 0> <row 1> ...<row 7> d
```

where: <row 0> is the 8-bit pattern for row 0

.
.
.

<row 7> is the 8-bit pattern for row 7

A simple checkerboard pattern would be defined as follows:

```
ESC * m 170 85 170 85 170 85 170 85 D
        |                          |
      Row 0                      Row 7
```

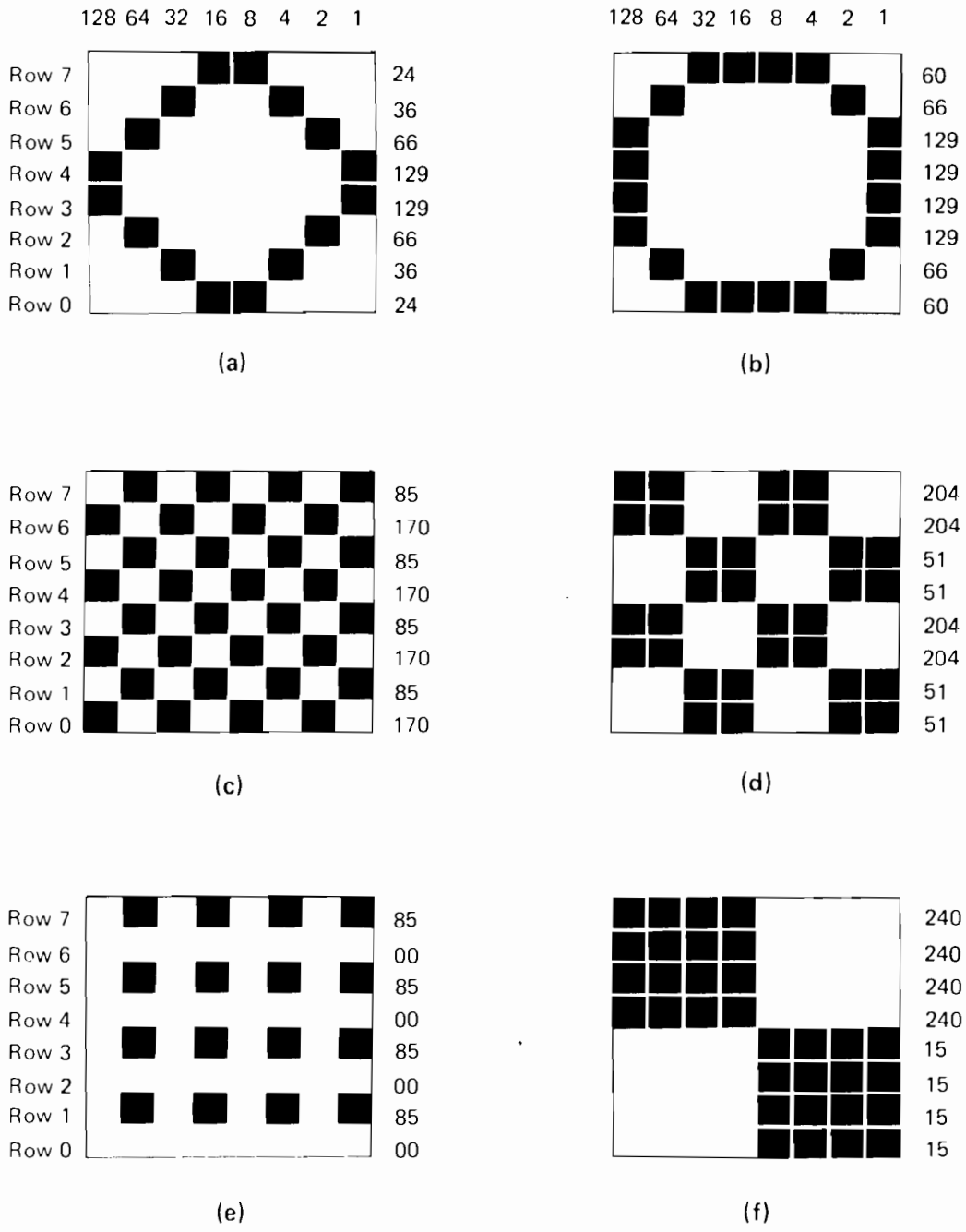


Figure 4-6. Area Pattern Examples

AREA FILL. A rectangular area can be filled with a pattern by simply sending the lower left (LL) and upper right (UR) coordinates of the rectangle. The coordinates can be in either absolute or relocatable format. The pattern used is selected by the Line Type command. This allows a choice of predefined and user defined line patterns as well as an 8×8 bit area pattern (refer to Define Area Pattern).

An easy way to selectively erase a portion of the graphics display would be to set the drawing mode to clear, select the solid vector line type (type = 1), and then use the area fill command to select the area to be cleared. Area fill is also useful for shading bar graphs or engineering drawings. The soft keys can be loaded with the proper escape sequences and then triggered to generate area patterns locally using either the current cursor or pen position as a coordinates.

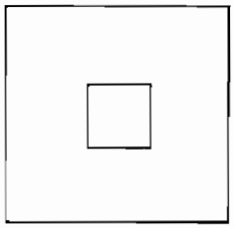
Area Fill, Absolute. The absolute area fill command uses the absolute coordinates of the area.

Area Fill, Absolute: `Et * m <XLL,YLL> <XUR,YLR> e`

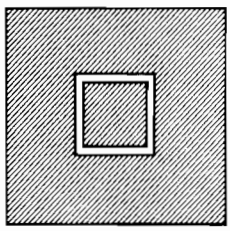
where: `<XLL,YLL>` and `<XUR,YUR>` are the absolute coordinates of the lower left and upper right corners of the area to be filled.

Example: Draw a box and then complement the entire graphics display. Note that repeating the `Et * m` sequence would restore the original display.

```
Et*p a f 150 150 200 150 200 200 150 200 150 150 Z
```



```
Et * m 3 a 1 b 0 0 719 359 E
```



Example: Clear the display area with lower left (LL) coordinates 0,0 and upper right (UR) coordinates 100,100.

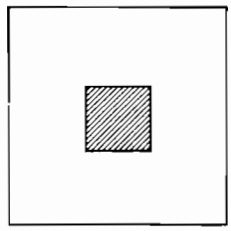
```
Et * m 1 a 1 b 0,0 100,100 E
```

clear mode — solid line — area to be cleared

If a predefined or user defined line pattern is selected, the area fill command can be used to provide area shading or complex patterns. If the user defined area pattern is selected, the area will be filled with the 8x8 area pattern (refer to Define Area Pattern).

Example: Using the area fill pattern shown in figure 4-6a, shade the area with XLL,YLL = 50,50 and XUR,YUR = 100,100.

```
Et * m 2 a 3 b 24 36 66 129 129 66 36 24 d 50 50 100 100 E
```



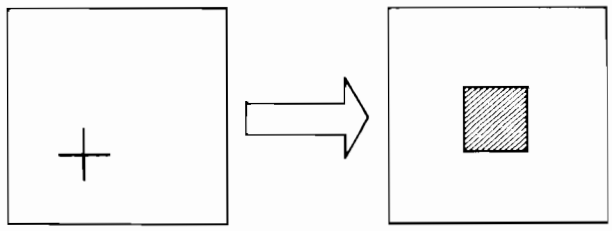
Area Fill, Relocatable. The relocatable area fill command uses area coordinates in relocatable ASCII format.

Area Fill, Relocatable: `Et * m <XLL,YLL> <XUR,YUR> f`

where: `<XLL,YLL>` and `<XUR,YUR>` are the relocatable coordinates for the lower left and upper right corners of the area to be filled.

Example: Using the area fill pattern shown in figure 4-6c, shade a 50× 50 unit area with the lower left corner at the current cursor position.

```
Et * m 2 a 3 b 85 170 85 170 85 170 85 170 d 1 0 0 50 50 F
```



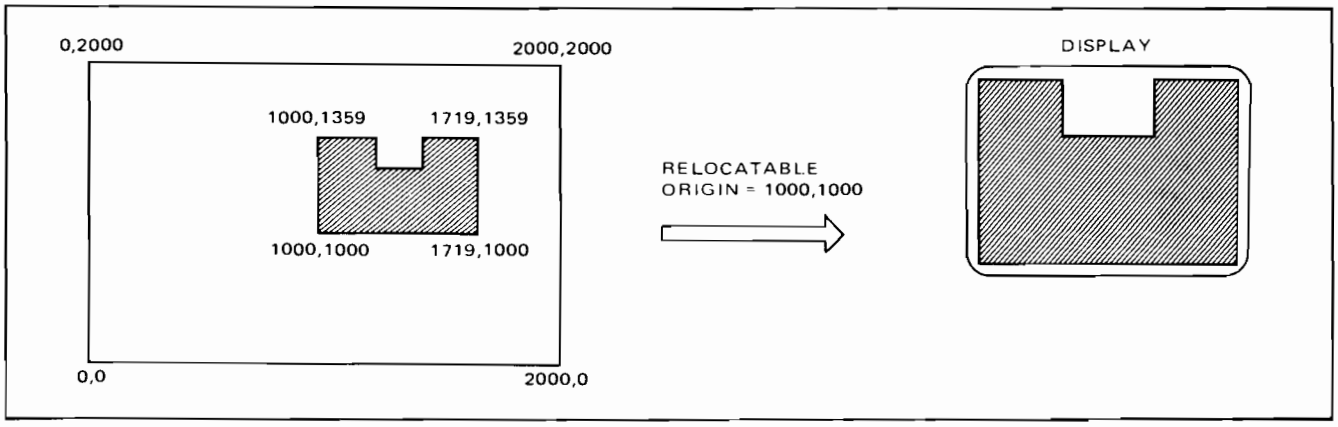


Figure 4-7. Relocatable Origin

RELOCATABLE ORIGIN. The relocatable origin allows you to use one set of data and drawing commands to display a figure at several different positions on the screen. (See the resistor example under ASCII Relocatable Format.)

You can also display portions of a figure that is too large to fit on the screen. You can create a "window" that can be positioned to display any 720 by 360 unit portion of the figure. The value of the relocatable origin is added to the relocatable data to obtain the coordinates used to draw the data. Figure 4-7 illustrates the effect of a Relocatable Origin on the display.

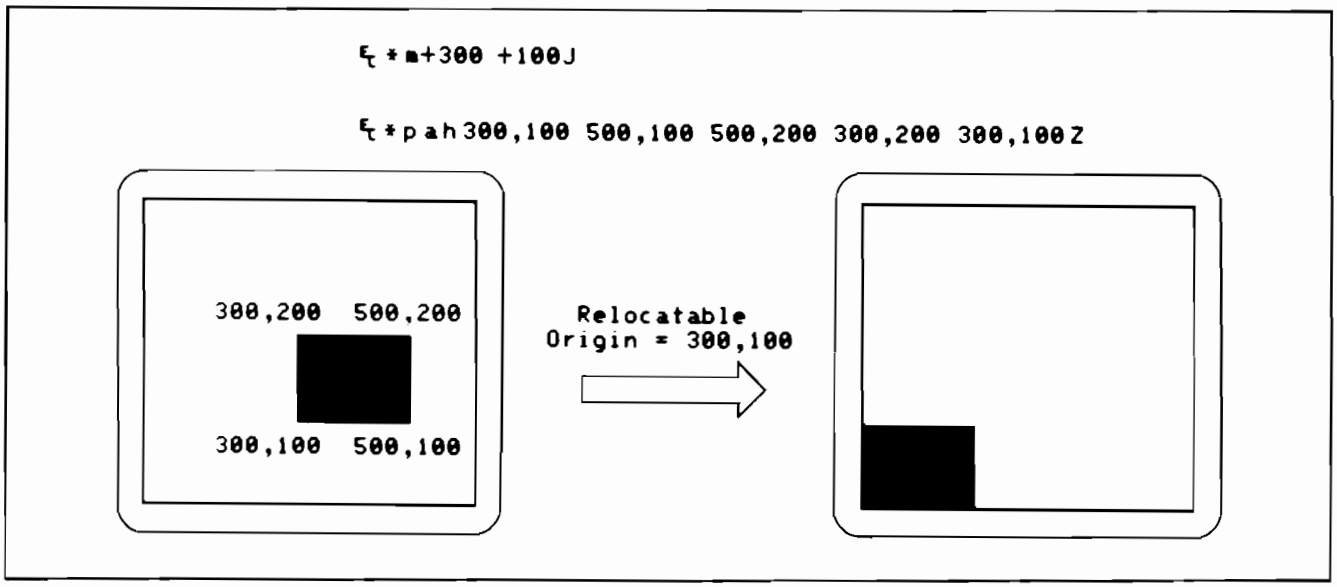
This technique eliminates the need to check boundary conditions or compute new data in order to display the desired portion of the figure. Simply set the relocatable origin to the proper value to display the desired portion of the figure and then send the unchanged figure data to the terminal. The terminal will then automatically select and adjust the "window" data.

Set Relocatable Origin Absolute. The relocatable origin can be set to any absolute coordinates using ASCII absolute format (−16384 to 16383).

Set Relocatable Origin Absolute: `Ec * m <X,Y> j`

where: `<X,Y>` are the x and y coordinates in ASCII absolute format.

Example: Set the relocatable origin to display the box in the figure so that the box is positioned at the lower left corner of the display.



Set Relocatable Origin to Current Pen Position. The relocatable origin can be set to the current pen position.

Set Relocatable Origin To Current Pen Position: ESC * m k

Set Relocatable Origin to Graphics Cursor Position. The relocatable origin can be set to the current graphics cursor position.

Set Relocatable Origin To Graphics Cursor Position: ESC * m l

RUBBER BAND LINE. The "rubber band" feature can be used to preview vectors before they are stored in the graphics memory or sent to the computer. This feature is normally used by the operator when inputting graphics data from the keyboard. When the rubber band feature is enabled, a temporary line is displayed in the current line type (refer to Line Type). The line is shown extending from the current pen position to the graphics cursor. The line is dynamic and will follow the cursor as it is moved. The line is not stored until a draw command is entered.

Rubber Band Line On: ESC * d m

Rubber Band Line Off: ESC * d n

SELECTING THE GRAPHICS DEFAULT PARAMETERS. Graphics parameters can be set to their default (power on or full reset) values. Table 4-5 lists the various parameters and their default values. Additional information can be found under the discussions of the individual parameters.

Set Graphics Default Parameters: ESC * m r

The current graphics mode and settings can be obtained with graphics status requests. Graphics status requests are described in the Status section. It may be desirable to reselect graphics settings before you send graphics data to the terminal.

Table 4-5. Graphics Parameter Default Values

PARAMETER	DEFAULT VALUE
Pen Condition	up
Line Type	0
Drawing Mode	set
Relocatable Origin	0,0
Text Size	1
Text Direction	1
Text Origin	1 (left, bottom justified)
Text Slant	0 (off)
Graphics Text	off
Graphics Video	on
Alphanumeric Video	on
Graphics Cursor	off
Alphanumeric Cursor	on
Rubber Band Line	off
Zoom	off
Zoom Size	1
Compatibility Mode	
Page Full Strap	1 (in)
GIN Strap	0 (CR only)

Plotting Sequences

All vector plotting sequences are initated by ESC * p. Table 4-6 lists the commands that can be used within a plotting sequence.

Table 4-6. Graphics Plotting Control Functions

ESC * p <parameters and data>	
PARAMETER	**DESCRIPTION**
a	lift the pen
b	lower the pen
c	use graphics cursor position as new point
d	draw a single dot at the current pen position
e	set relocatable origin = current pen position
f	use ASCII absolute format
g	use ASCII incremental format
h	use ASCII relocatable format
i	use binary absolute format
j	use binary short incremental format
k	use binary incremental format
l	use binary relocatable format
z	NOP/synch

After ESC * p has been sent, the drawing format is normally specified before data is sent.

If no format is specified, ASCII absolute is assumed. There is no explicit draw vector command. When enough parameter bytes to specify a single end point have been received (the number depends on the format used), the pen is moved from its current position to the new end point. (See figure 4-8.) If the pen is down, a vector will be drawn. If

the pen is up, the pen is moved to the new point (without drawing a vector) and lowered. The new end point becomes the current pen position.

Note that if a parameter byte is lost or garbled in transmission, all following end points will be improperly read. To minimize data errors caused by the loss of a data byte, any command can be used to reset the parameter count and restore synchronization. Nops (z), redundant format, or pen down commands can also be inserted to insure synchronization if necessary.

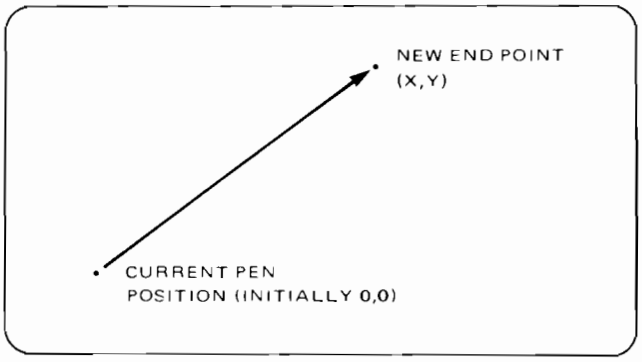


Figure 4-8. Current Pen Position And New End Point

Graphics sequences can extend indefinitely. In general, longer sequences are preferred as they minimize the overhead necessary for a plot sequence. ESC * p <format> must be sent for each series of vectors. As the sequence length decreases, the percentage of preamble characters increases, and the vector drawing rate goes down. The worst possible case would be to send ESC * p <format> for each vector; approximately 50% of the characters sent would be overhead, reducing vector speed by a factor of 2.

The general format for an absolute plotting sequence is:

```
Esc * p i a <byte1> <byte2> <byte3> <byte4>(z)
<byte1> <byte2> <byte3> <byte4> ...
...<byte1> <byte2> <byte3> <byte4>Z (or any
capital command)
```

Each block of 4 bytes specifies a single point. The "i" indicates that absolute format is to be used. The "a" raises the pen before it is moved to the point specified by the next four bytes and lowered. A NOP (z) can be added to insure synchronization, if necessary. The lowered pen draws a vector as it moves to the next point, and so on. The capital "Z" terminates the plotting sequence.

The vector end point formats allow the pen to be moved completely off the screen (an absolute coordinate of 1000, for example). The actual range of the pen position can be from −16384 to 16383. Vectors that extend beyond the screen are clipped so that they will not wrap around.

PEN CONTROL. The terminal uses the concept of a "pen" in drawing vector data. The pen can be lifted or lowered as well as be positioned using absolute or relative coordinates. For example, the pen is lifted, moved to a starting coordinate, lowered and moved to an endpoint to draw a line. The pen is initially in the up state and positioned at absolute coordinates 0,0 following power up or a full reset. If the pen is raised and coordinates given, the pen is moved to the coordinates and then lowered. The pen is normally left in the down position.

Raise Pen: Esc * p a

Lower Pen: Esc * p b

VECTORS. Graphic data is made up of vectors. Each vector is specified by the current graphic starting point and an end point. The current graphic starting point is one of the following:

0,0 Initial starting point

Last point defined by the user with the MOVE key

Last point defined by the user with the DRAW key

Last point defined by the graphics cursor (ESC * p c)

Last point defined by data in a draw or move command (ESC * p f/g/h/i/j/k/l)

Graphic points are specified in one of following formats:

- ASCII Absolute
- ASCII Incremental
- ASCII Relocatable
- Binary Absolute
- Binary Incremental
- Binary Short Incremental
- Binary Relocatable

If no format is specified in the graphic command, ASCII absolute format is assumed. More than one point can be given in a command. This minimizes communications overhead. Tables 4-7, 4-8, and 4-9 provide a reference for computing data bytes used in the various vector formats.

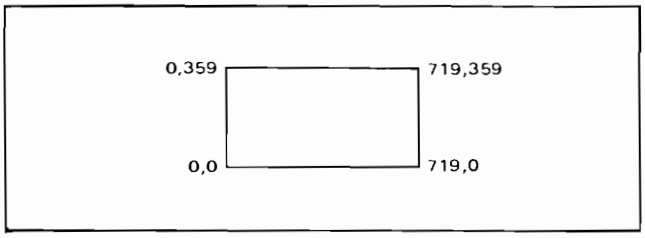
ASCII FORMATS. In the ASCII formats, coordinates are specified with ASCII characters 0 through 9. This means that numeric characters generated by a simple print statement can be used to specify X,Y pairs. The first value is used as the X coordinate, and the second as the Y coordinate.

Spaces or commas must be used to delimit the X and Y values. Excess delimiters are ignored. Digits following a decimal point are ignored (i.e. 123.456 is read as 123).

Exponential notation cannot be used. Consequently, the values must be in integer form. The number of bytes necessary to specify a single end point depends on the magnitude of the values.

ASCII Absolute Format. The values used in the ASCII absolute format can range between −16384 and 16383. Note that only points where X is in the range 0 to 719 and Y is in the range 0 to 359 will be visible on the screen. The following example draws vectors around the perimeter of the screen.

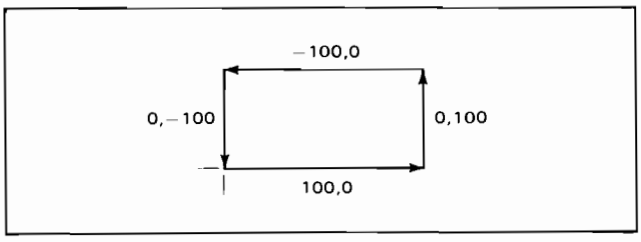
```
Et * p a 0,0 719,0 719,359,0,359,0,0Z
```



Since no format is indicated, ASCII absolute is assumed. The "a" raises the pen, which is moved to (0,0) and lowered. Vectors are then drawn to (719,0), (719,359), (0,359), and back to (0,0). (Note that the values are delimited by spaces or commas. The capital Z (a nop) terminates the sequence. Imbedded carriage return and line feed characters are ignored.

ASCII Incremental Format. In the ASCII incremental format you can specify a delta X and a delta Y. These values are added to the current pen position to obtain a new end point. The first value is read as delta X and the second as delta Y. For example to draw a square 100 units on a side, the following sequence could be used:

```
Et * p g 100 0 0 100 -100 0 0 -100 Z
```



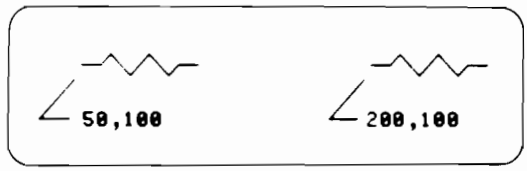
Beginning at the current pen position, a series of vectors is drawn by moving the pen 100 units to the right, up 100 units, left 100 units, and finally down 100 units. The same figure could have beeen drawn at any screen location by first positioning the pen to the desired starting point before sending the drawing sequence.

ASCII Relocatable Format. The ASCII relocatable format allows you to use a relocatable origin to be added to the incoming X and Y coordinate values. The resultant values are then treated as absolute coordinates by the terminal. The relocatable format allows you to use absolute data as if it were incremental by merely changing the relocatable origin. For example, symbol elements specified in absolute coordinates can be drawn in different locations as shown in the following example.

Example: Draw a resistor symbol stored in absolute coordinates at screen locations 50,100 and 200,100.

```
Resistor Data = 0,10
                10,10
                15,15      0,20    60,0
                25,5
                35,15
                45,5
                50,10      0,0     60,0
                60,10

Et*m50,100J
Et*pah0,10 10,10 15,15 25,5 35,15
                45,5 50,10 60,10Z
Et*m200,100J
Et*pah0,10 10,10 15,15 25,5 35,15
                45,5 50,10 60,10Z
```



BINARY FORMATS. In binary format all points are sent in a packed binary format. The coordinate values are sent using the bit patterns of the ASCII characters listed in table 4-7. The number of characters required to specify a coordinate depends on the format used. The values for X and Y coordinates can be from −16384 to 16383.

Binary Absolute Format. Binary absolute data is plotted with respect to an origin at 0,0. Four bytes are required to specify a single end point. A 10 bit coordinate in the range 0-1023, is sent for both x and y.

The bytes are ordered as follows:

BIT	7	6	5	4	3	2	1	
BYTE 1	0	1	X9	X8	X7	X6	X5	HI X
BYTE 2	0	1	X4	X3	X2	X1	X0	LOW X
BYTE 3	0	1	Y9	Y8	Y7	Y6	Y5	HI Y
BYTE 4	0	1	Y4	Y3	Y2	Y1	Y0	LOW Y

Although it is possible to send coordinates in the range 0 to 1023, only points in the range 0-719 for X, and 0-359 for Y are visible on the screen. Vectors going off the screen are clipped. If the data requires scaling, this must be done before the data is sent to the terminal.

The following example shows how the 4 data bytes are computed. The numbers are converted to the 10 bit binary equivalent. Bits 7 and 6 are set to 01 to indicate a parameter.

```
X = 0 = 00000 00000        Y = 0 = 00000 00000
        HI X  LOW X                HI Y  LOW Y

            BYTE 1 = 01 00000 = SPACE HI X
            BYTE 2 = 01 00000 = SPACE LOW X
            BYTE 3 = 01 00000 = SPACE HI Y
            BYTE 4 = 01 00000 = SPACE LOW Y

X = 360 = 01011 01000      Y = 180 = 00101 10100
          HI X  LOW X                HI Y  LOW Y

              BYTE 1 = 01 01011 = +  HI X
              BYTE 2 = 01 01000 = (  LOW X
              BYTE 3 = 01 00101 = %  HI Y
              BYTE 4 = 01 10100 = 4  LOW Y
```

An escape sequence to draw a vector from 0,0 to 360,180 is as follows:

```
ESC * p i a SP  SP  SP  SP  +  (  %  4  Z
            \X=0/   \Y=0/ \X=360/ \Y=180/
```

ESC * p selects a plotting sequence. The "i" specifies absolute format. The "a" raises the pen up. The first 4 bytes (all spaces) move the raised pen to 0,0, where it is lowered. The next 4 bytes specify the point 360,180. After the 4th byte is received, the pen is moved to that point, drawing a vector. The capital "Z" terminates the escape sequence. Note that if spaces are used in the data sequence they are interpreted as data resulting in an improper plot.

Binary Short Incremental Format. The short incremental format uses two bytes to specify a delta X and a delta Y in the range −16 to +15. The five least significant bits are interpreted as a signed, two's complement number. This number is added to the current pen position to obtain the new end point. The data bytes are ordered as follows:

```
   BIT    7   6   5   4   3   2   1
BYTE 1    0   1   <     DELTA X     >
BYTE 2    0   1   <     DELTA Y     >
```

The following example illustrates the computation and use of the short incremental format:

```
DELTA X = -12 = 10100    DELTA Y = 6 = 00110
BYTE1 = 01 10100 = 4  DELTA X
BYTE2 = 01 00110 = &  DELTA Y
```

The following sequence moves the pen to 360,180 in absolute format, then draws a vector to X = 360−12 = 350, y = 180+6 = 186.

```
                    absolute
ESC * p i a + ( % 4 j 4 & <byte1><byte2>...Z
Short Incremental   Short Incremental vectors
```

Binary Incremental Format. Incremental is similar to short incremental, but with a larger range. Using six bytes, delta X and Y can range from −16384 to +16383.

```
   BIT 7 6  5    4    3    2    1

BYTE1 0 1 DX14 DX13 DX12 DX11 DX10 HI  DELTA X
BYTE2 0 1 DX9  DX8  DX7  DX6  DX5  MID DELTA X
BYTE3 0 1 DX4  DX3  DX2  DX1  DX0  LOW DELTA X

BYTE4 0 1 DY14 DY13 DY12 DY11 DY10 HI  DELTA Y
BYTE5 0 1 DY9  DY8  DY7  DY6  DY5  MID DELTA Y
BYTE6 0 1 DY4  DY3  DY2  DY1  DY0  LOW DELTA Y
```

The following example shows how incremental data bytes are generated.

```
DELTA X = -400 = 11111  10011   10000
                 HI DX  MID DX  LO DX

DELTA Y =  100 = 00000  00011   00100
                 HI DY  MID DY  LO DY

BYTE 1 = 01 11111 = ?      HI  DELTA X
BYTE 2 = 01 10011 = 3      MID DELTA X
BYTE 3 = 01 10000 = 0      LO  DELTA X

BYTE 4 = 01 00000 = space  HI  DELTA Y
BYTE 5 = 01 01001 = #      MID DELTA Y
BYTE 6 = 01 00100 = $      LO  DELTA Y
```

Table 4-7. Characters Used in Packed Data Formats

ASCII Character	Bit Pattern	ASCII Character	Bit Pattern
SP	01 0 0000	0	01 1 0000
!	01 0 0001	1	01 1 0001
"	01 0 0010	2	01 1 0010
#	01 0 0011	3	01 1 0011
$	01 0 0100	4	01 1 0100
%	01 0 0101	5	01 1 0101
&	01 0 0110	6	01 1 0110
'	01 0 0111	7	01 1 0111
(	01 0 1000	8	01 1 1000
)	01 0 1001	9	01 1 1001
*	01 0 1010	:	01 1 1010
+	01 0 1011	;	01 1 1011
,	01 0 1100	<	01 1 1100
-	01 0 1101	=	01 1 1101
.	01 0 1110	>	01 1 1110
/	01 0 1111	?	01 1 1111

Binary Relocatable Format. Binary relocatable format specifies absolute X and Y coordinates in the range −16384 to +16383 using 6 bytes. The value specified in the relocatable origin command is taken to be the 0,0 point. The actual screen address is computed by the terminal by adding the relocatable origin to the X,Y pair.

BIT	7	6	5	4	3	2	1	
BYTE 1	0	1	X14	X13	X12	X11	X10	HI X
BYTE 2	0	1	X9	X8	X7	X6	X5	MID X
BYTE 3	0	1	X4	X3	X2	X1	X0	LOW X
BYTE 4	0	1	Y14	Y13	Y12	Y11	Y10	HI Y
BYTE 5	0	1	Y9	Y8	Y7	Y6	Y5	MID Y
BYTE 6	0	1	Y4	Y3	Y2	Y1	Y0	LOW Y

The following example shows how relocatable data bytes are computed.

```
RELOC X = -600 = 11111 01101 01000
                 HI X  MID X LOW X

RELOC Y =  200 = 00000 00110 01000
                 HI Y  MID Y LOW Y

BYTE 1 = 01 11111 = ?      HI  X
BYTE 2 = 01 01101 = -      MID X
BYTE 3 = 01 01000 = (      LOW X

BYTE 4 = 01 00000 = space  HI  Y
BYTE 5 = 01 00110 = &      MID Y
BYTE 6 = 01 01000 = (      LOW Y
```

Table 4-8. Absolute Format Addressing Bytes

	0	1	2	3	4	5	6	7	8	9
0	■■	■!	■"	■#	■$	■%	■&	■'	■(	■)
10	■*	■+	■,	■-	■.	■/	■0	■1	■2	■3
20	■4	■5	■6	■7	■8	■9	■:	■;	■<	■=
30	■>	■?	!■	!!	!"	!#	!$	!%	!&	!'
40	!(	!)	!*	!+	!,	!-	!.	!/	!0	!1
50	!2	!3	!4	!5	!6	!7	!8	!9	!:	!;
60	!<	!=	!>	!?	"■	"!	""	"#	"$	"%
70	"&	"'	"(	")	"*	"+	",	"-	".	"/
80	"0	"1	"2	"3	"4	"5	"6	"7	"8	"9
90	":	";	"<	"=	">	"?	#■	#!	#"	##
100	#$	#%	#&	#'	#(	#)	#*	#+	#,	#-
110	#.	#/	#0	#1	#2	#3	#4	#5	#6	#7
120	#8	#9	#:	#;	#<	#=	#>	#?	$■	$!
130	$"	$#	$$	$%	$&	$'	$(	$)	$*	$+
140	$,	$-	$.	$/	$0	$1	$2	$3	$4	$5
150	$6	$7	$8	$9	$:	$;	$<	$=	$>	$?
160	%■	%!	%"	%#	%$	%%	%&	%'	%(	%)
170	%*	%+	%,	%-	%.	%/	%0	%1	%2	%3
180	%4	%5	%6	%7	%8	%9	%:	%;	%<	%=
190	%>	%?	&■	&!	&"	&#	&$	&%	&&	&'
200	&(	&)	&*	&+	&,	&-	&.	&/	&0	&1
210	&2	&3	&4	&5	&6	&7	&8	&9	&:	&;
220	&<	&=	&>	&?	'■	'!	'"	'#	'$	'%
230	'&	''	'(	')	'*	'+	',	'-	'.	'/
240	'0	'1	'2	'3	'4	'5	'6	'7	'8	'9
250	':	';	'<	'=	'>	'?	(■	(!	("	(#
260	($	(%	(&	('	((	()	(*	(+	(,	(-
270	(.	(/	(0	(1	(2	(3	(4	(5	(6	(7
280	(8	(9	(:	(;	(<	(=	(>	(?	)■	)!
290	)"	)#	)$	)%	)&	)'	)(	))	)*	)+
300	),	)-	).	)/	)0	)1	)2	)3	)4	)5
310	)6	)7	)8	)9	):	);	)<	)=	)>	)?
320	*■	*!	*"	*#	*$	*%	*&	*'	*(	*)
330	**	*+	*,	*-	*.	*/	*0	*1	*2	*3
340	*4	*5	*6	*7	*8	*9	*:	*;	*<	*=

	0	1	2	3	4	5	6	7	8	9
350	*>	*?	+■	+!	+"	+#	+$	+%	+&	+'
360	+(	+)	+*	++	+,	+-	+.	+/	+0	+1
370	+2	+3	+4	+5	+6	+7	+8	+9	+:	+;
380	+<	+=	+>	+?	,■	,!	,"	,#	,$	,%
390	,&	,'	,(	,)	,*	,+	,,	,-	,.	,/
400	,0	,1	,2	,3	,4	,5	,6	,7	,8	,9
410	,:	,;	,<	,=	,>	,?	-■	-!	-"	-#
420	-$	-%	-&	-'	-(	-)	-*	-+	-,	--
430	-.	-/	-0	-1	-2	-3	-4	-5	-6	-7
440	-8	-9	-:	-;	-<	-=	->	-?	.■	.!
450	."	.#	.$	.%	.&	.'	.(	.)	.*	.+
460	.,	.-	..	./	.0	.1	.2	.3	.4	.5
470	.6	.7	.8	.9	.:	.;	.<	.=	.>	.?
480	/■	/!	/"	/#	/$	/%	/&	/'	/(	/)
490	/*	/+	/,	/-	/.	//	/0	/1	/2	/3
500	/4	/5	/6	/7	/8	/9	/:	/;	/<	/=
510	/>	/?	0■	0!	0"	0#	0$	0%	0&	0'
520	0(	0)	0*	0+	0,	0-	0.	0/	00	01
530	02	03	04	05	06	07	08	09	0:	0;
540	<■	0=	0>	0?	1■	1!	1"	1#	1$	1%
550	1&	1'	1(	1)	1*	1+	1,	1-	1.	1/
560	10	11	12	13	14	15	16	17	18	19
570	1:	1;	1<	1=	1>	1?	2■	2!	2"	2#
580	2$	2%	2&	2'	2(	2)	2*	2+	2,	2-
590	2.	2/	20	21	22	23	24	25	26	27
600	28	29	2:	2;	2<	2=	2>	2?	3■	3!
610	3"	3#	3$	3%	3&	3'	3(	3)	3*	3+
620	3,	3-	3.	3/	30	31	32	33	34	35
630	36	37	38	39	3:	3;	3<	3=	3>	3?
640	4■	4!	4"	4#	4$	4%	4&	4'	4(	4)
650	4*	4+	4,	4-	4.	4/	40	41	42	43
660	44	45	46	47	48	49	4:	4;	4<	4=
670	4>	4?	5■	5!	5"	5#	5$	5%	5&	5'
680	5(	5)	5*	5+	5,	5-	5.	5/	50	51
690	52	53	54	55	56	57	58	59	5:	5;
700	5<	5=	5>	5?	6■	6!	6"	6#	6$	6%
710	6&	6'	6(	6)	6*	6+	6,	6-	6.	6/

Note: ■ indicates a "space" character; every coordinate address must consist of the two characters shown in the table.

Table 4-9. Incremental (Short) Vector Bytes

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?

Recording Graphics Functions

The [DISPLAY FUNCTIONS] key can be used to display and record the graphics escape sequences or the action of graphics control keys. The control sequences are entered into the alphanumeric display each time a command is executed. The sequences can then be stored on cartridge tape using Edit Mode or a Record or Copy command. Table 4-10 lists the graphics control sequences that are generated when DISPLAY FUNCTIONS is on.

Table 4-10. Graphics Control Sequences Used In Record Operations

Key	Sequence	Description
[U] [D] [L] [R]	none	Graphics cursor controls
[CURSOR FAST]	none	Graphics cursor fast
[SHIFT] [RB LN]	Ec * dM Ec * dN	
[STOP]	Ec * aB Ec * dT	
[ZOOM IN]	Ec * dnI	"n" is the new zoom size
[ZOOM OUT]	Ec * dnI	"n" is the new zoom size
[ZOOM]	Ec * dH Ec * dG	
[G CURSOR]	Ec * dL Ec * dK	
[SHIFT] [G DSP]	Ec * dC Ec * dD	
[SHIFT] [CLEAR]	Ec * dA	
[SHIFT] [A D DSP]	Ec * dF Ec * dE	
[SHIFT] [TEXT]	Ec * dS Ec * dK Ec * m4A	Turns on cursor, jam pattern (if not in scaled Compatibility Mode), and turns on Graphics Text Mode.
[SHIFT] [T SZE]	Ec * mnM	"n" is the text size
[SHIFT] [T ANG]	Ec * mnN Ec * mO Ec * mnN Ec * mP	"n" is the text angle

Note that the DRAW and MOVE commands do not execute unless the graphics cursor is on.

Figure 4-9 shows the sequences generated when drawing a simple box. The graphics cursor is initially on and positioned at 0,0.

Raster Data Transfers

Image data can be transferred directly from the graphics memory to disc, compatible printer (such as the HP 2631G or HP 7245A) or datacomm. In addition, a disc or a remote CPU can transfer image data directly to the graphics memory. The following paragraphs discuss the data format for such transfers, as well as additional escape sequences that can be used to control the placement of image data on the terminal's display.

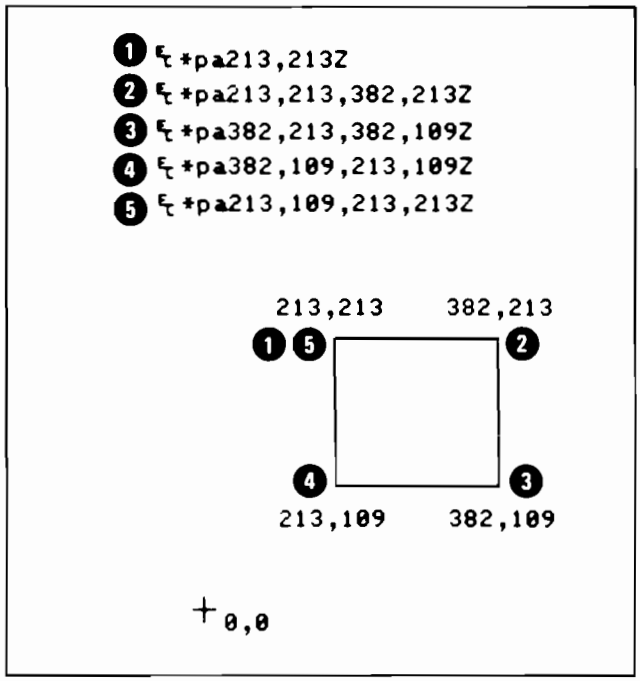


Figure 4-9. Recording Graphics Sequences

RASTER OUTPUT FORMAT. The terminal always outputs raster data in the same format, regardless of the destination. This format consists of a start record, 360 data records, and an ending record. The start record initializes the tranfer and may dimension the display. Data records consist of a byte count followed by the proper number of binary data bytes. The data bytes contain the data for one image line, 720 dots, scanned left to right. (See figure 4-10.) The display is scanned top to bottom, so that the first data record represents the top-most raster line of the display. A data record will either contain 90 bytes (720 dots) or, in the case of a blank raster line, will not contain any data. The ending record terminates the transfer. Escape sequence preambles are used to differentiate the different types of records.

When transferring the raster data into the graphics memory, the current drawing mode is used. Jam mode copies the data exactly, complement mode complements the memory, etc.

The S and T parameters inform the receiving device of the size of the picture. Some devices can use this information to center the picture. If the destination device is the datacomm, and the G and H straps on the Processor PCA are closed (DC1 handshake with standard datacomm), a DC1 control character must be sent by the receiving device on a record by record basis to initiate the transfer of each record. That is, each time the receiving CPU sends DC1, one raster record will be sent. If this handshake is not enabled, the entire display is sent without stopping.

SENDING IMAGE DATA FROM CPU TO THE TERMINAL. Using a format similar to the example above, a remote CPU can send image data to the terminal. ESC * r A is first sent to initialize the terminal for a raster transfer. Before the binary data for each data record is sent, an ENQ/ACK handshake must be performed as follows, to allow time for the terminal to go into binary mode.

1. Send ESC * b 90 W ENQ.
2. When the terminal sends back ACK, send the 90 data bytes.

This handshake is similar to that used when writing binary data to tape. If no data is sent (ESC * b 0 W for a blank line) the ENQ/ACK handshake need not be done. Fewer than 90 bytes can be sent. As each line is sent, it is drawn below the previous line, so the data must be sent left to right, top to bottom. Commands described in the next paragraph can be used to position the image anywhere on the screen.

Positioning Commands. When the terminal copies the graphics memory to tape or other destination, the entire graphics memory is always transferred. When sending data to the terminal, however, it is possible to transfer less than a full screen. The data sent can be positioned anywhere on the CRT. It is also possible to select and display only a subset of the data sent to the terminal ("windowing"). (See figure 4-13.)

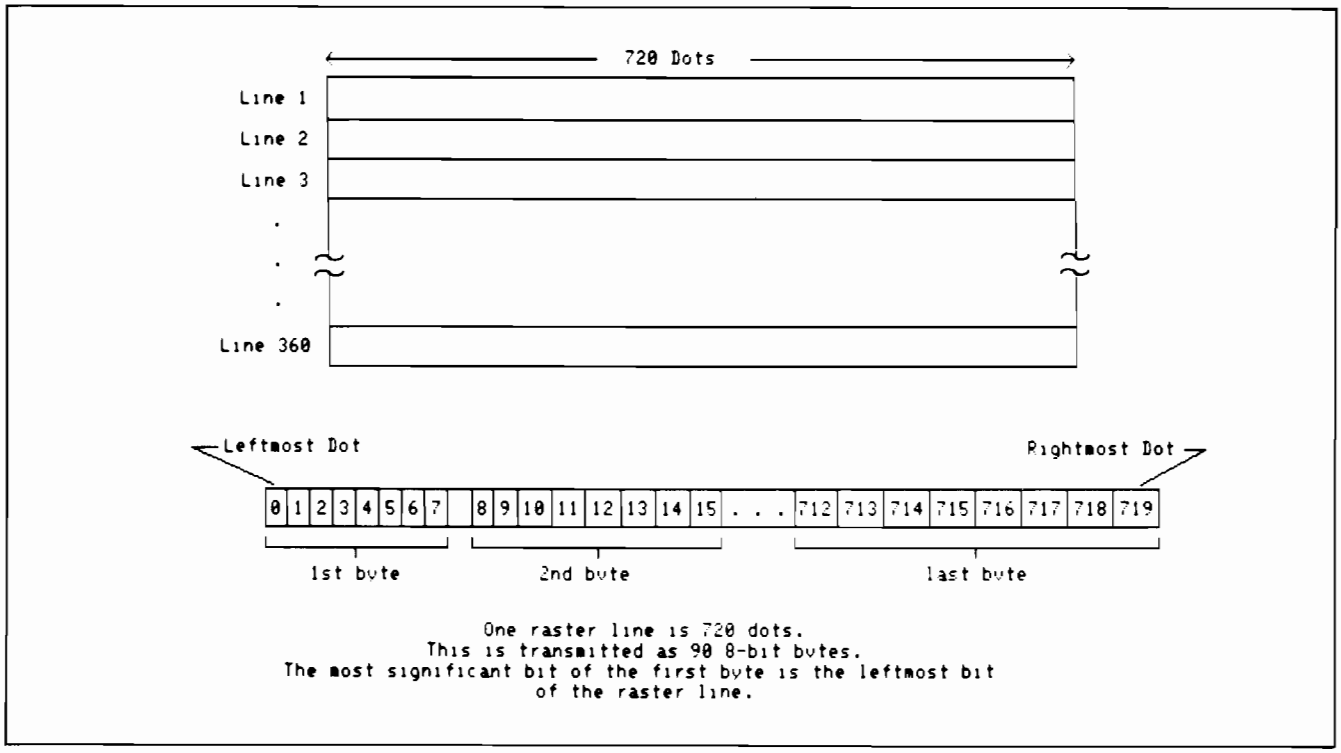


Figure 4-10. Raster Data Format

X and Y Offsets. Ordinarily, the starting point for drawing raster data is the upper left corner of the screen. That is, when Esc * r A is received, the pen position is initialized to that point. It is possible to offset that point as follows:

Esc * r <x offset> x <y offset> y A

For example, if it was desired to move the starting point down 100 dots from the top, and in 40 dots from the left side (see figure 4-11), then the following sequence would be used to initialize the transfer:

Esc * r 40x 100y A

Note that the 'A' command must follow these or any other windowing commands. Also, the X offset is truncated to a multiple of 8 dots. An X offset of 50, for example, would be interpreted as an offset of 48.

If 90 bytes (720 dots) were sent in the above example, each raster line would exceed the right side of the screen by 40 dots. This overflow is ignored.

An example to illustrate the transfer of a small raster image to a specific point on the screen is shown in figure 4-12.

Windowing Commands. It is possible to set up the terminal to ignore specific raster lines and data bytes before a transfer is initiated. This allows the user to window out a selected portion of the data.

These escape sequences can be entered locally or can be sent before the raster transfer begins. The commands are as follows:

Esc * r

<horizontal window address> m	Gives the number of dots to ignore at the start of each line.
<vertical window address > n	Gives the number of raster lines to skip at the start of the picture.
<horiz. window dimension > p	Gives the number of dots to draw in each line.
<vert. window dimension > q	Gives the total number of raster lines to draw.

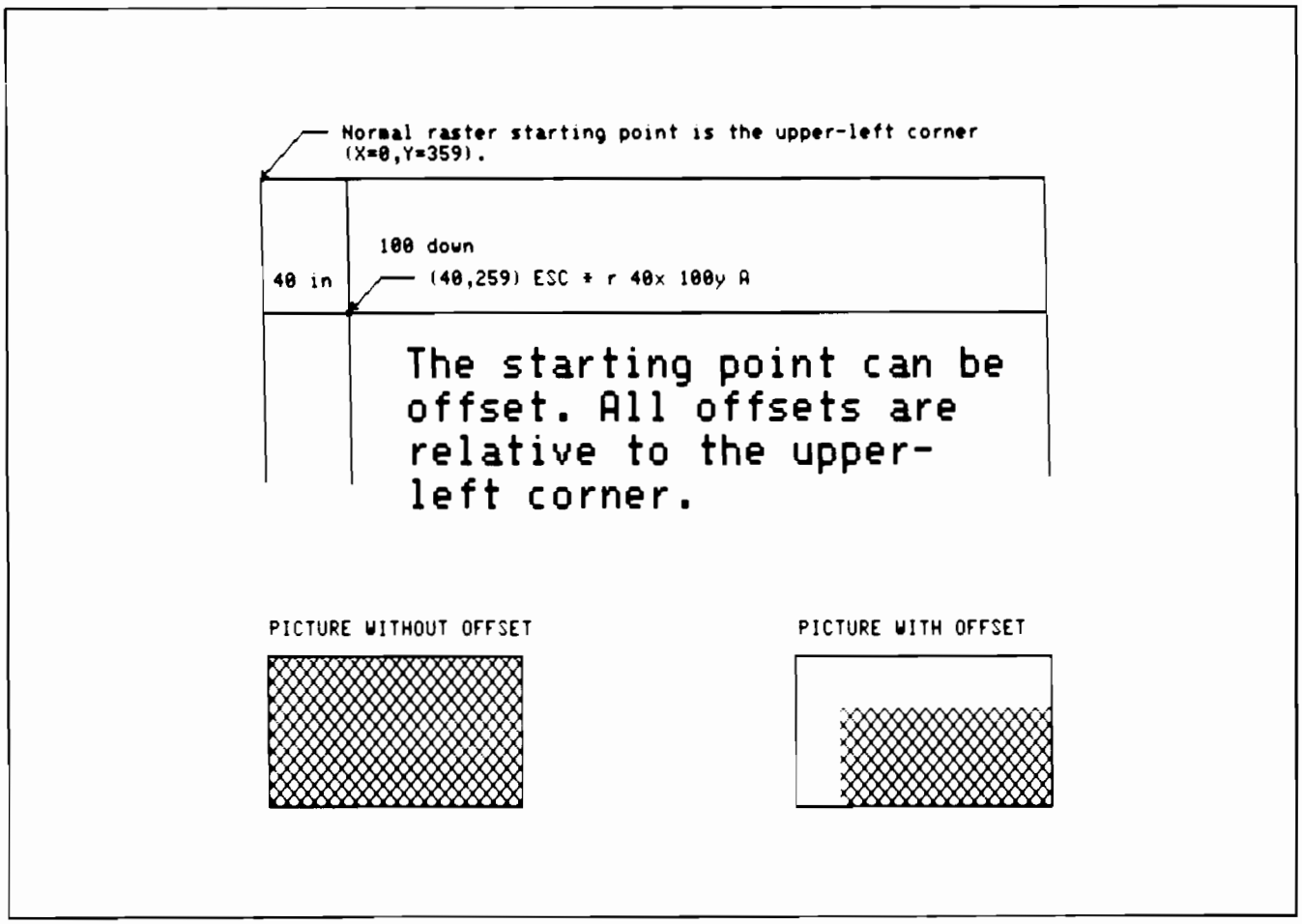


Figure 4-11. Raster Data X and Y Offsets

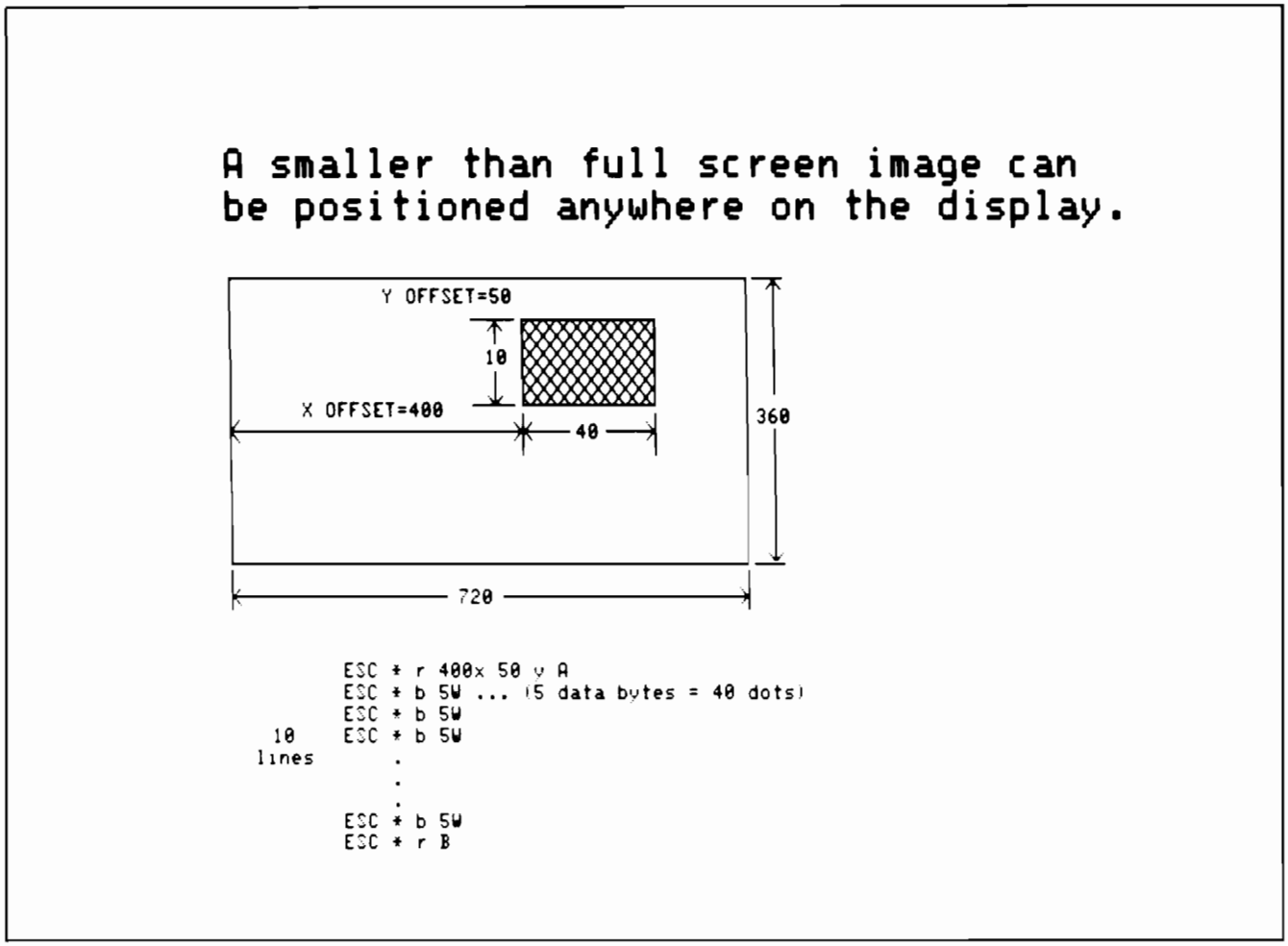


Figure 4-12. Raster Data Positioning

These commands will be explained by the use of an example (see figure 4-13). The escape sequence Esc * r 40m 10n 400p 1000Q would be interpreted as follows:

1. The first 10 raster lines would be ignored (10N).

2. Only the next 100 raster lines would be displayed (100Q). Any succeeding data would be ignored until the next raster dump started, as signaled by Esc * r A.

3. On the 100 raster lines which would be drawn, the first 40 dots (40M) would be ignored.

4. After the first 40 dots are skipped, only the next 400 dots are drawn (400P). Any data on the raster line after that is ignored.

The result of this would be to display a 400 by 100 dot subset of the original 720 by 360 picture. The P and Q commands select the size of the subset, and the M and N commands determine where the subset is located in the original data.

As with the X offset parameter, the X window address (M) and the X window size (P) are truncated to the nearest multiple of 8 dots.

Offset and window parameters stay in effect for succeeding pictures until they are explicitly changed or the terminal is (hard) reset.

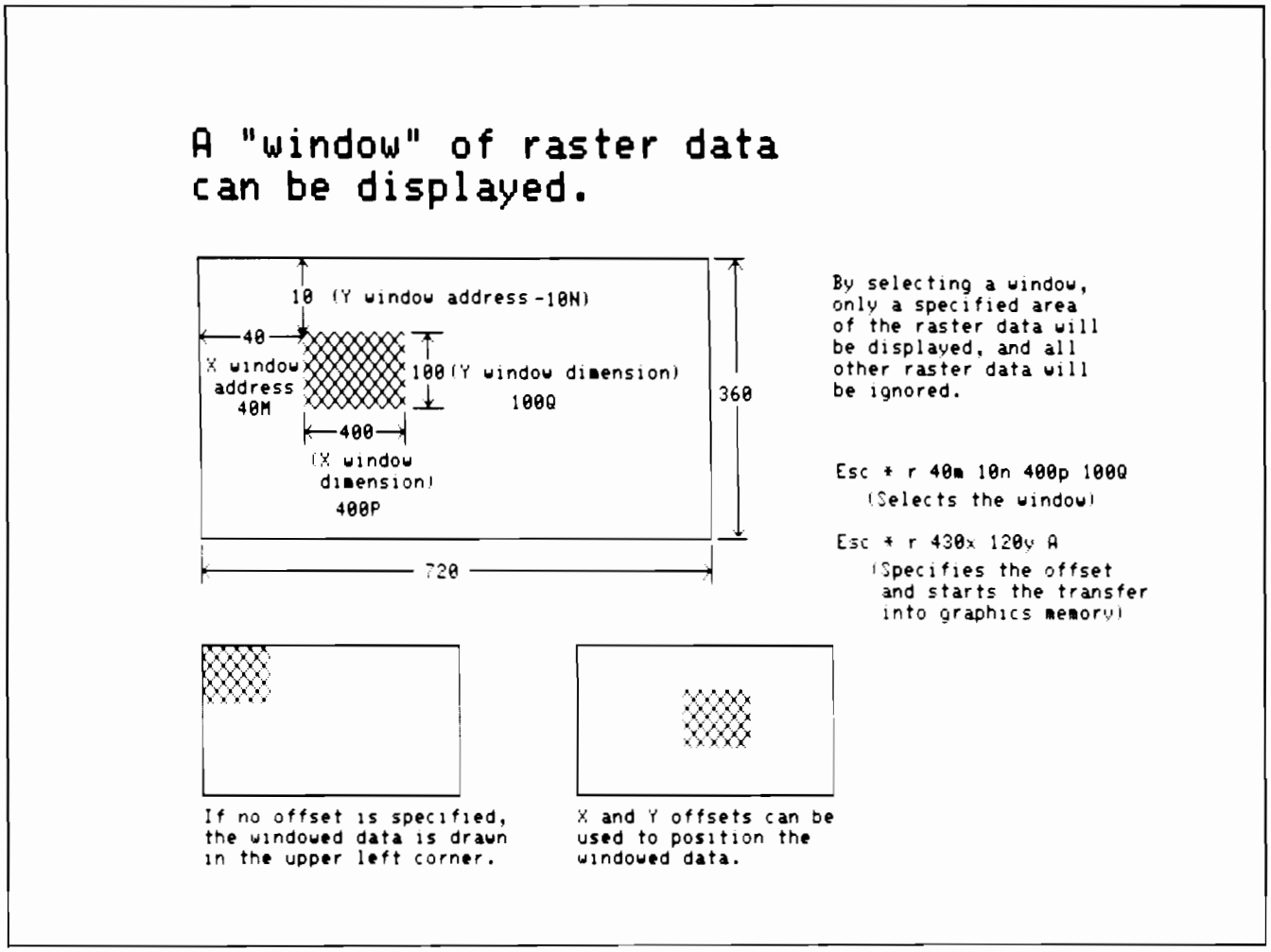


Figure 4-13. Raster Data Windowing

READING THE GRAPHICS MEMORY. A remote CPU can read the data stored in the terminal's graphics memory. To read the entire display, the command sequence "TRANSFER ALL FROM GRAPHICS TO DATACOMM" can be used. It is also possible to read a subset of the display, as follows:

1. An X and a Y offset are specified to select the upper left corner of the area to be read.
2. The sequence Esc * r A is sent to initialize the terminal for reading.
3. The sequence Esc * b <count> R (DC1) is sent to read the desired number of data bytes in each scan line.

Example:

The CPU sends Esc * r 80x 100y A to select the area to be read.

The CPU sends Esc * b 10 R to read 10 bytes (80 dots).

The terminal sends back 10 bytes.

The CPU sends Esc * b 10 R to read 10 bytes in the next (lower) scan line, etc.

SUMMARY OF COMMANDS. As with all parameterized sequences, as many lower case commands as desired can be concentrated together. An upper case command terminates the sequence. All of the commands in table 4-11 begin with Esc * r.

Any raster formatting parameters should be placed before the "A" in the sequence; otherwise, the current parameters will be in effect when the "A" (prepare for raster dump) is executed.

THIS	NOT THIS
Esc*r 10m 15n 525p 50q A	Esc*r a 10m 15n 525p 50Q

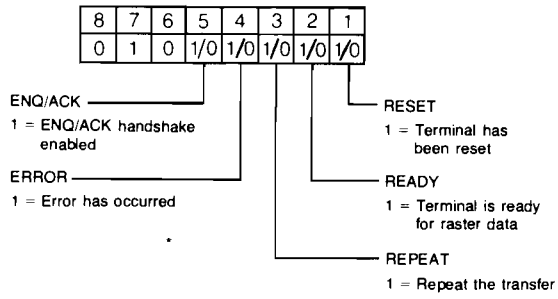
Graphics Hardcopy Operations

An HP 2631G Printer or HP 7245A Plotter Printer can print the contents of graphics memory. The printer is interfaced to the terminal using the ROM/HP-IB PCA. Procedures for configuring and installing this accessory are contained in the Installation section. The printer must be assigned as the "Destination" device then the raster data is transferred as described previously (refer to "Raster Data Transfers").

A video hardcopy subsystem is also available to make printed copies of the graphics display. The video hardcopy subsystem uses the HP 13254A Video Interface. Instructions for installing and configuring the interface are given in the Installation section. Procedures for making video copies are given in the Commands section. If the interface is configured as address 04 (refer to the Configurations section and the *13254A Installation and Service Manual,* part no. 13254-90001), a PRINT command (`Mark File Header on External Printer`) can be entered locally at the terminal or sent from a computer. Copies can also be initiated manually from the hard copy unit itself.

Table 4-11. Summary of Raster Dump Commands

Esc * r Sequence	Description
A	Prepare for raster dump.
B	End of raster dump.
C	Erase screen.
D	Turn video on.
E	Return raster status. Upon receipt of DC1 (if G and H closed), the terminal returns one byte of status, interpreted as follows:
F	Not used.
G	Not used.
H	Not used.
I	Set all parameters to default value.
J	Return raster size status. The terminal returns the string '720,360'.
K	Return model number. The string 2647F is returned.
L	Not used.
M	Horizontal window address.
N	Vertical window address.
O	Not used.
P	Window horizontal dimension.
Q	Window vertical dimension.
R	Not used.
S	Horizontal size of picture.
T	Vertical size of picture.
U	Not used.
V	Not used.
W	Not used.
X	X offset.
Y	Y offset.
Z	NOP.



Graphics Text

Text strings can be written directly into the graphics image memory. An internal character generator converts the ASCII codes into a dot matrix representation which is drawn as vectors. The character set includes upper and lower case (96 characters) and will be drawn as a 5 by 7 matrix in a 7 by 10 cell, with descenders for lower case. This character set is in addition to the normal alphanumeric character set. While this character set may seem redundant, it offers the following advantages:

- Characters can be drawn at any dot position, rather than the 24 by 80 alphanumeric character positions.
- Characters can be rotated in multiples of 90 degrees.
- Characters can be scaled in size, from 1 to 8 times.
- Characters can be slanted 45 degrees for an italics-like effect.
- Lines of characters can be right, left, or center justified.
- In zoom mode, characters in the graphics memory are magnified.

Figure 4-14 shows the graphics character set.

KEYBOARD CONTROL OF GRAPHICS TEXT. Graphics text can be entered directly from the keyboard. The backspace, carriage return, and line feed functions work as expected (even on inverted text), making it easy to add or edit titles and labels. A summary of keyboard operations affecting Graphics Text Mode is given in table 4-12.

Table 4-12. Graphics Text Keyboard Functions

KEY	DESCRIPTION
TEXT	Selects the graphics image memory as the destination for all text. Characters entering from the keyboard, datacomm, or disc are drawn as vectors in the graphics memory using the current text size and angle (see the T SZE and T ANG keys). The drawing mode is initially set to jam pattern to allow for backspacing and retyping of characters. The graphics cursor indicates the position of the next character. Moving the graphics cursor will cause the next text line to begin at the new cursor position. The carriage return, line feed, and backspace functions work normally.
STOP	Terminates Text mode.
T SZE	Increases the character size from 1 to 8X. The smallest character is a 5 by 7 matrix in a 7 by 10 cell. Increasing the size makes the dots bigger while the character is still drawn as a 5 by 7 matrix.
T ANG	Sets the character orientation (multiples of 90 degrees) and turns slant on or off.
TAB	Spaces one graphics text character to the right. (The actual direction of movement will depend on the text orientation.)
CNTL K	(Vertical Tab). Spaces one graphics text line up. (The actual direction of movement will depend on the text orientation.)
In addition, the following keys function in the same manner as for alphanumeric text characters: RETURN, LINE FEED, BACKSPACE, ←, →, ↑, ↓	

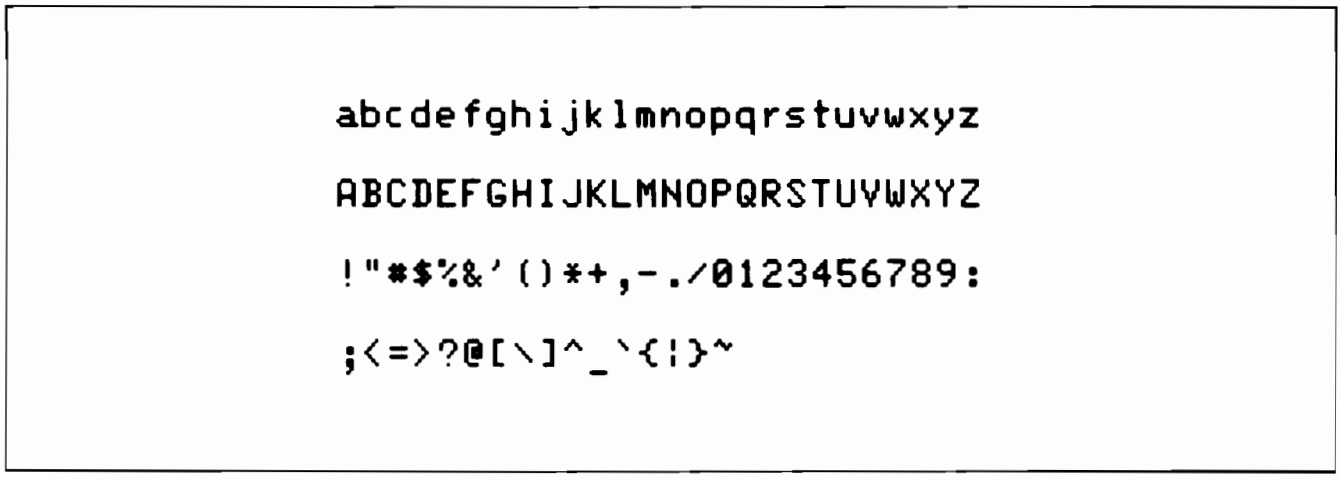


Figure 4-14. Graphics Text Characters

PROGRAM CONTROL OF GRAPHICS TEXT. All of the parameters for graphics text can be set programmatically. Commands are of the form: Esc * m <parameter> <command>. The command can be alone or as part of another Esc * m sequence.

Size. The ASCII characters 1 through 8 specify the character size for graphics text. A "1" indicates the smallest character, a 5 by 7 dot matrix character in a 7 by 10 cell. Increasing the size increases the size of the dots. If a text size of 1 is specified, each dot in the cell is one dot on the screen. A size of 2 uses 4 screen dots for each character dot (2 X 2), and so on (see figure 4-15). A size of "1" is the default.

Set Graphics Text Size: Esc * m <size> m

Text Direction. This command uses the ASCII characters 1 through 4 to specify the text orientation (see figure 4-16). This also changes the direction of line feed, carriage return, and backspace.

1 — Normal (upright, the default)
2 — Rotated 90 degrees counter clockwise
3 — Rotated 180 degrees counterclockwise (inverted)
4 — Rotated 270 degrees counter clockwise

Set Graphics Text Orientation: Esc * m <orientation> n

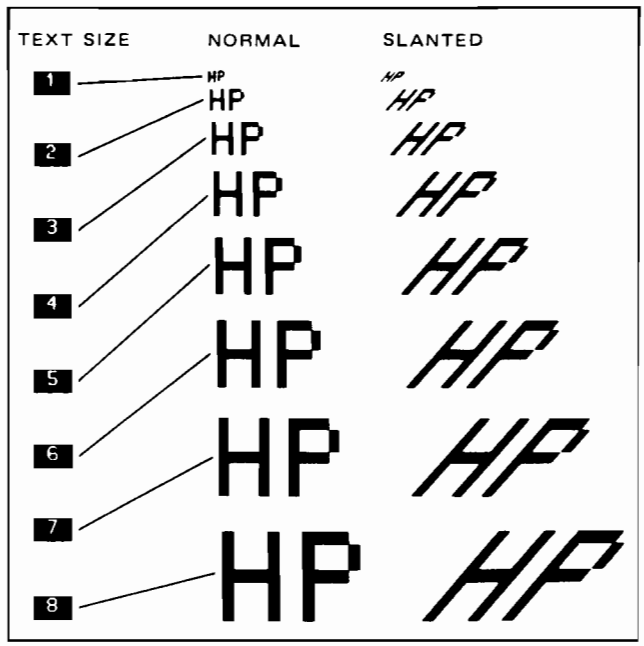


Figure 4-15. Graphics Text Sizes

Slant. The graphics text characters can be slanted 45 degrees for an italics effect.

Turn On Graphics Text Slant: Esc * m o

Turn Off Graphics Text Slant: Esc * m p

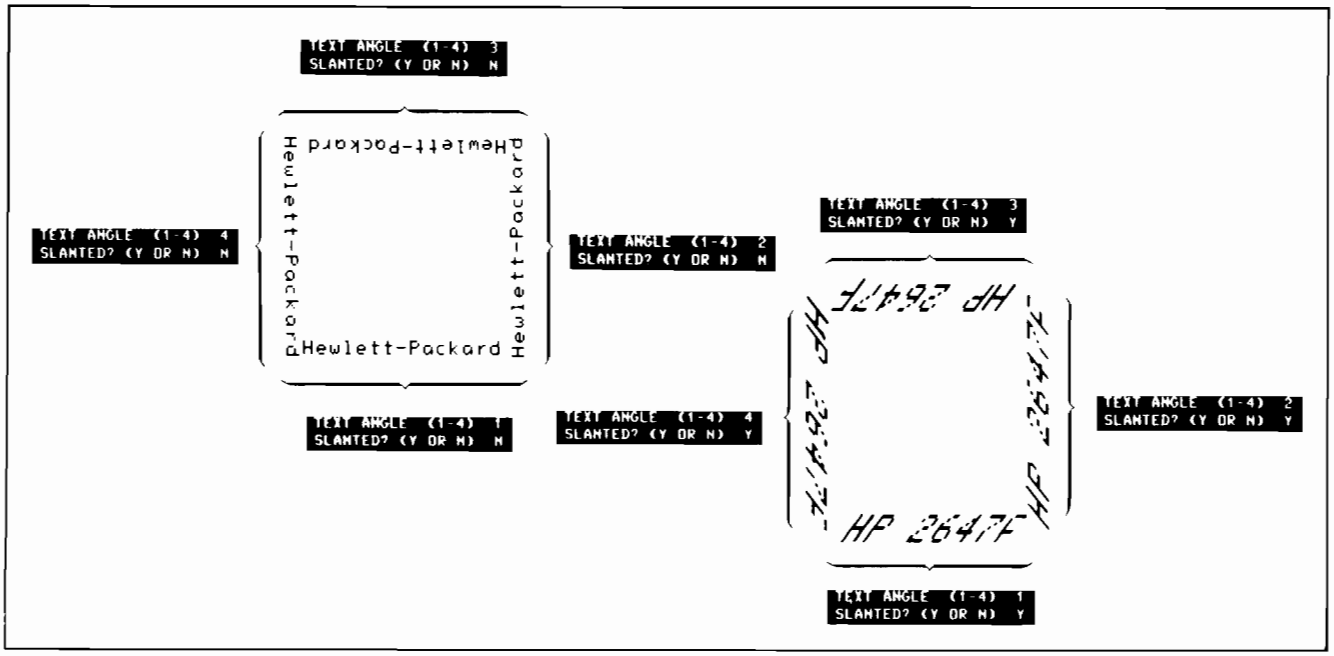
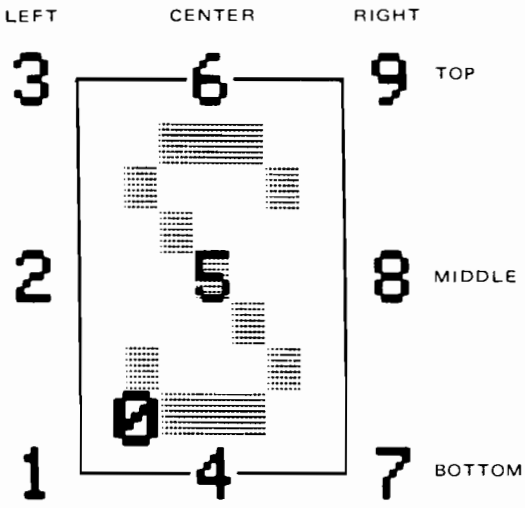


Figure 4-16. Graphics Text Direction

Justification/Origin. Text strings can be automatically right or left justified, or centered about a specified point. An ASCII character 0 through 9 indicates the origin (justification and base line) for characters with respect to the current pen position. This function is useful when drawing labels. (Refer to the Label command.)

Set Graphics Text Justification: `Esc * m <origin> q`

If text is left justified, the current pen position is the left margin. Center causes the label to be centered on the pen position. Right justify selects the pen position as the right margin. Bottom, middle, and top select the base line for the line of text.



The numbers 1-9 represent the cursor position with respect to the character cell used for graphics text characters. The number 0 represents the cursor position with respect to the character (not this cell).

For example, if text was to be right justified and set with a base line on top of the normal character position, the number "9" would be used. Figure 4-17 illustrates the various text positions.

When centering or right justification is used, the text strings are buffered (stored) until all of the characters in the string have been received. The string end is detected by a CR or LF. The string is not displayed until the CR or LF is received. This may be confusing when entering text from the keyboard. The maximum length of a string when center or right justifying is 80 characters (not including the CR(LF)). In all cases, data written beyond the edge of the screen is lost. There is no automatic RETURN when the screen boundry is reached.

Turning Graphics Text On and Off. Graphics text mode can be turned on or off from a program. These two commands use the `Esc * d` sequence but are discussed here under graphics text for completeness.

On. This command will cause Graphics Text Mode to be turned on. All displayable characters will be stored in the graphics memory. If the command is entered from the keyboard using the [SHIFT] [TEXT] keys the graphics cursor is turned on to indicate the position where the next character will be displayed. The drawing mode is initially set to jam mode to permit overstrike replacement of characters. A difference mode, such as set or complement, can be selected at any time.

Text is drawn using the current text assignments for size and orientation. Graphics text mode accepts CR, LF, BS, HT, and VT as control characters. The [L], [R], [U], and [D] keys can be used to position the graphics cursor in character increments.

Turn On Graphics Text Mode: `Esc * d s`

If the graphics cursor is moved or a DRAW or MOVE command is executed, the graphics text margin is moved to the new cursor or pen position.

Characters are drawn using the current drawing mode (set, clear, or jam). If set mode is used, entering a character, backspacing, and entering a second character causes an overstrike. If jam mode is used, the new character will replace the old character.

If a lower case "s" is used, additional escape parameters can be appended to the sequence. Otherwise the next characters will be routed to the graphics memory.

Examples:

```
Esc * d s k 100,100 o B
```
— set graphics memory
— position cursor at 100,100
— turn on cursor

```
Esc * d S  This is a text string^R
```

Off. This sequence turns off graphics text mode and restores normal alphanumeric operation.

Turn Off Graphics Text Mode: `Esc * d t`

Graphics Text Status. You can check the current text settings with a graphics text status request. Refer to the Status section for additional information.

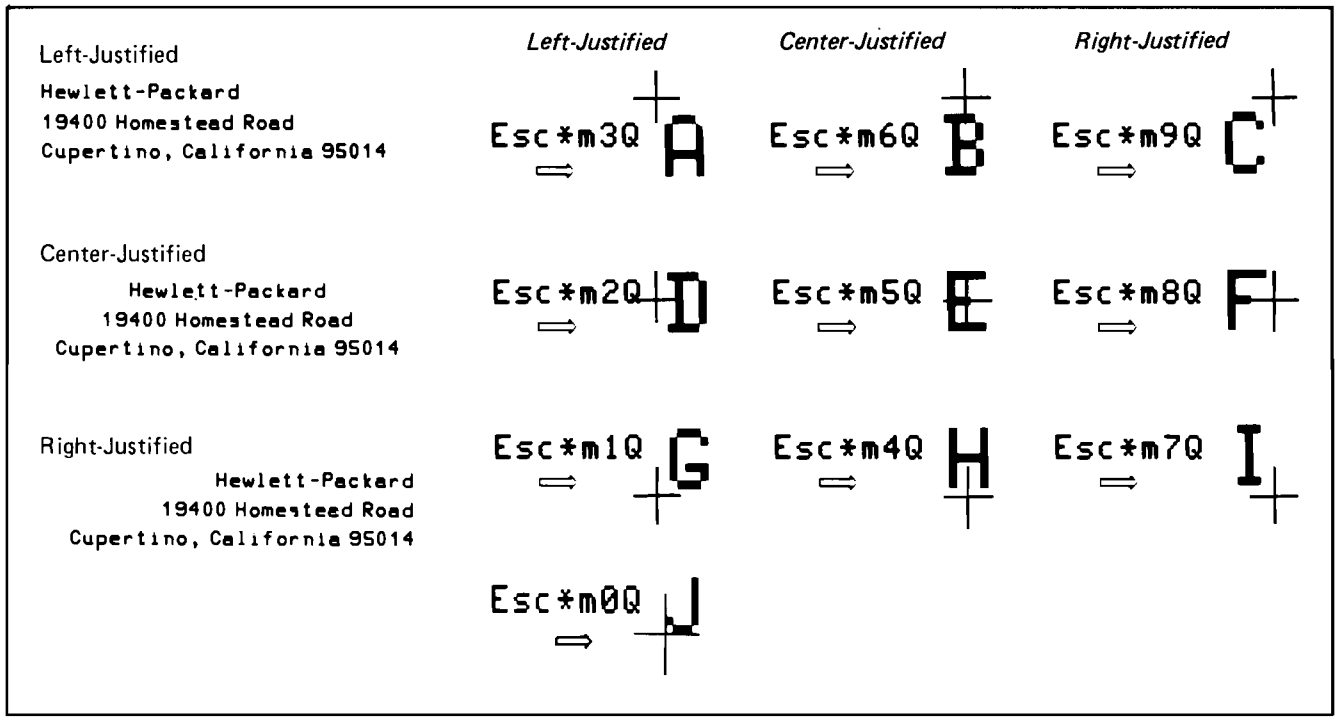


Figure 4-17. Graphics Text Justification

Label. This sequence is used to send a single record of graphics text to the terminal. The characters are stored in the graphics memory using the current text size, angle, slant, and justification. The label is drawn beginning at the current pen position.

Graphics Text Label: ESC * 1 <text string> ^R (LF)

The record must end with a CR, LF, or both. A CR moves the pen to its original position when the label command was first received. An LF moves the pen down one line (character spacing). Note that the actual directions moved following a CR or LF depend on the text orientation selected.

The maximum record length is 80 characters, not including the ESC * 1 preamble or the CR(LF).

Example: ESC*1This is a sample label^R LF

Compatibility Mode

Compatibility Mode allows the terminal to plot data intended for a terminal using a display with 1024 by 1024 addressable points. This mode makes it possible to use graphics programs developed for use with other graphics terminals with a minimum of reprogramming.

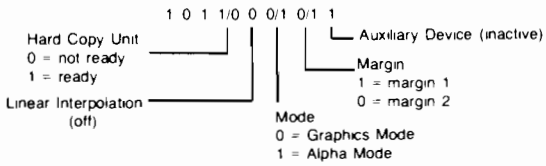
The terminal operates in two submodes while in Compatibility Mode. In Alphanumeric mode the terminal simply displays alphanumeric data on the screen as in normal operation. In Graphics mode the terminal responds to alphanumeric data as vector coordinates. Normally the terminal will be switched between these modes to display messages, plot graphics figures, and then display additional messages. These modes are controlled with several control sequences. (These sequences are ignored or acted on differently if the terminal is not set for Compatibility Mode.) Table 4-13 lists the terminal's responses to Compatibility Mode control sequences.

If delays are required, the baud rate can be lowered or fill characters added to prevent data loss when operating the terminal at high speeds. Refer to the Data Communications section.

Vectors are drawn using the current line type and line drawing mode. This gives you the capability of drawing dotted and dashed lines, etc. by changing the program to send the additional escape sequences. In general, all of the normal features of the terminal (display enhancements, tape control, etc.) are available only in the Alphanumeric mode.

Compatibility Mode is turned on by selecting either scaled or unscaled operation. Escape sequences controlling Compatibility Mode begin with ESC*t. This preamble is then followed with one or more commands. These commands are listed in table 4-14. As in all other escape sequences, a capital letter ends the sequence. Figure 4-18 contains examples of typical escape sequences.

Table 4-13. Compatibility Mode Control Sequences

CONTROL SEQUENCE	DESCRIPTION	RESPONSE
ESC ENQ	Read status and alpha cursor position	<status byte><HI X><LO X> <HI Y><LO Y><terminator>
	 The terminal will return one of the following characters as the status byte: 1 - Margin 2, Graphics Mode 3 - Margin 1, Graphics Mode 5 - Margin 2, Alpha Mode 7 - Margin 1, Alpha Mode	
ESC ETB (20 ms delay) ESC ENQ	Read graphics cursor position	<HI X><LO X><HI Y><LO Y><terminator>
ESC ETB	Read graphics cursor position when key struck	<KEY><HI X><LO X><HI Y><LO Y><terminator>
ESC EOT	Make hardcopy	
ESC FF	End graphics mode, clear screen, and home cursor	
GS	Go into graphics mode (draw vectors)	
US	Go into alpha mode	
BS	Backspace (H^c). Moves 1 space left (14 units)	
HT	Horizontal Tab (I^c). Moves 1 space right (14 units)	
CAN	End graphics mode	
LF	Line Feed (J^c). Moves 1 line down (22 units)	
VT	Vertical Tab (K^c). Moves 1 line up (22 units)	

NOTES

The terminal will normally respond with an XOFF character when an ENQ character is received. Compatibility Mode disables the terminal's ENQ/XOFF handshake. Compatibility Mode causes most control codes to be ignored.

The Read Status, alpha cursor position, and graphic cursor position cause block transfers to the computer systems. If the computer system does not use the DC1/DC2 handshake, straps G and H on the Processor PCA must be OPEN for these transfers to occur. (Refer to "Multicharacter Transfers" in the Commands section.

COMPATIBILITY MODE STRAPS. Compatibility Mode operation is controlled by Keyboard Interface switches P and Q. These switches can be set manually or programmatically using the Esc. . . sequence. The P and Q switches determine the terminal's mode of operation after being initialized (power up or full reset). The switches are interpreted as follows:

SWITCHES		DESCRIPTION
(Open=1, Closed=0)		
P	Q	
0	0	Normal graphics operation
0	1	Unscaled Compatibility Mode (expanded data comm buffer)[1]
1	0	Scaled Compatibility Mode (expanded data comm buffer)[1]
1	1	Normal graphics operation (expanded data comm buffer)[1]

[1] To obtain the larger buffer, the P and Q switches must be set physically. Refer to Section V.

In addition, when in Compatibility Mode, you can select the following optional capabilities:

Graphic Input Terminator. You can select the terminator sent by the terminal following the input of cursor address information. The terminator can be a CR, CR and EOT, or no terminator.

Page Full Busy. When this strap is in, the keyboard will be locked after the 35th line of text is received from the computer. The terminal can be cleared by pressing SHIFT CLEAR. This strap is ignored in Unscaled Mode.

Page Full Break. When this strap is in, the terminal will send a 200ms break signal to the computer after the 35th line of text is displayed. The terminal may also be set to BUSY (see Page Full Busy). When out, the strap will cause the cursor to home and the next 35 lines of text to be set with a left margin at x = 256. This strap is ignored in Unscaled Mode.

The commands to control these strap options are listed in table 4-14. Refer to the manual for the replaced graphics terminal for additional information on the operation of these straps and how they should be set.

GRAPHIC DATA. There are differences in display size (720 × 360 for the HP 2647F versus 1024 × 780 for other terminals) and line length (24 lines of 80 characters for the HP 2647F versus 35 lines of 74 characters for other terminals). See figure 4-19.

Table 4-14. Commands for Selecting Compatibility Mode

COMMAND	CODE
TURN SCALED COMPATIBILITY MODE ON (P open)	Esc & s 1 p 0 Q
TURN UNSCALED COMPATIBILITY MODE ON (Q open)	Esc & s 0 p 1 Q
TURN COMPATIBILITY MODE OFF (P,Q closed)	Esc & s 0 p 0 Q
The following commands simulate straps used on other graphics terminals:	
SET GRAPHICS INPUT TERMINATOR STRAP 0 — Carriage return only (Normal position) 1 — Carriage return and EOT 2 — No carriage return, no EOT	Esc * t <byte1> a
SET PAGE FULL BREAK STRAP 0 — Out (Normal position) 1 — In	Esc * t <byte1> b
SET PAGE FULL BUSY STRAP 0 — Out 1 — In (Normal position)	Esc * t <byte1> c
NOP	z

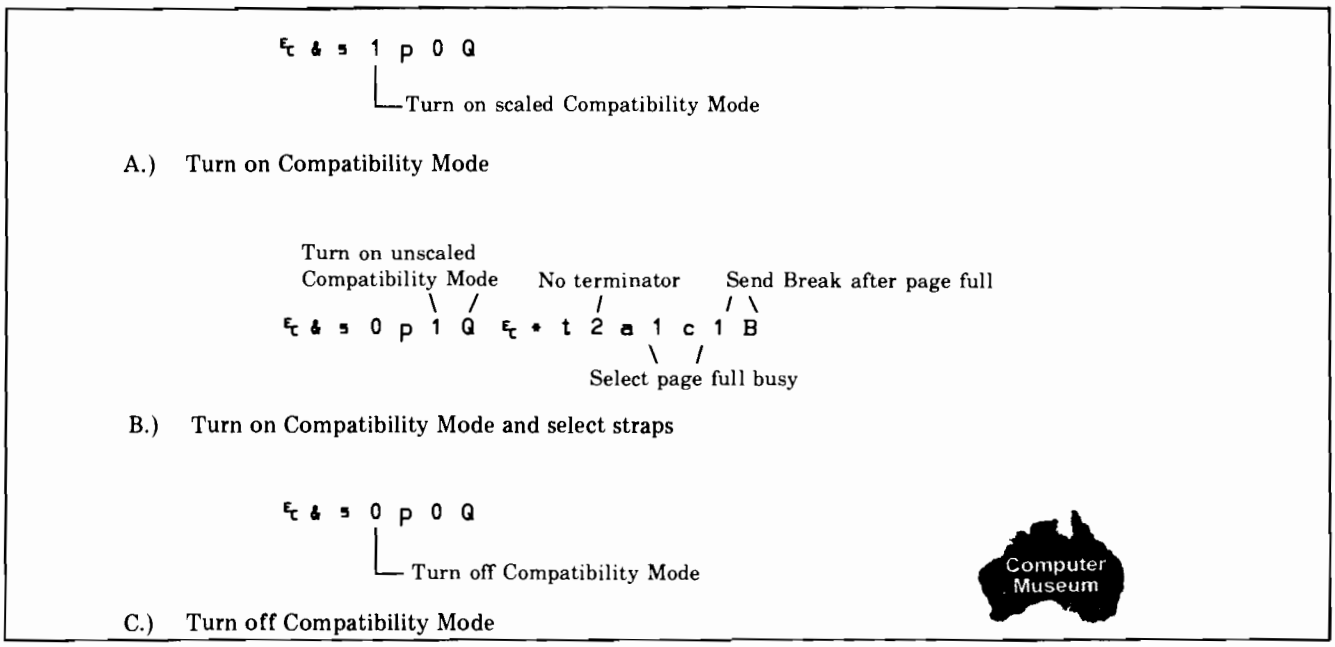


Figure 4-18. Turning on Compatibility Mode

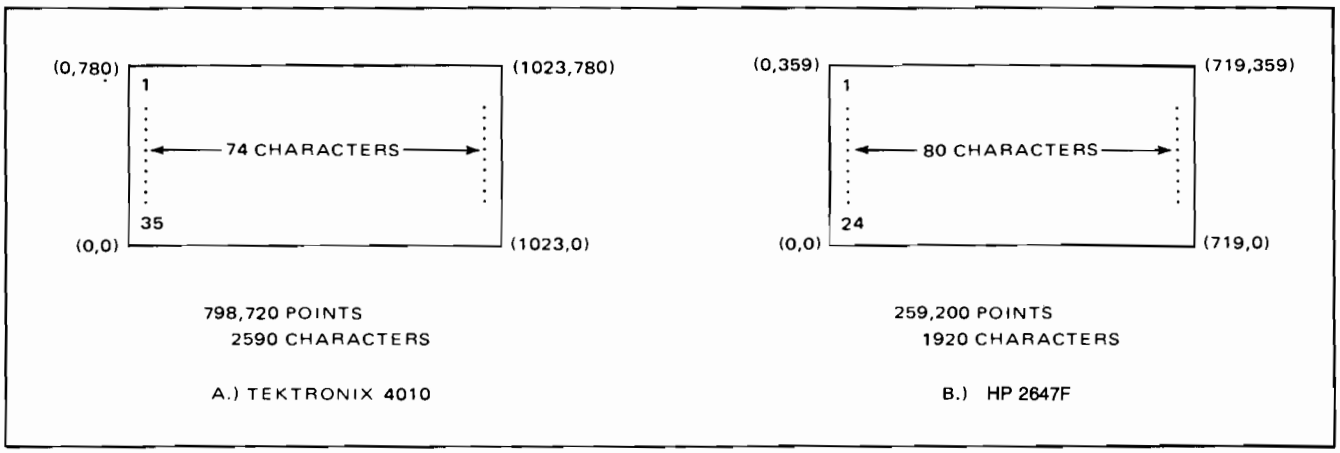


Figure 4-19. Comparison of a Terminal with 1024 × 780 Display and the HP 2647F

Graphic data can be drawn either scaled or unscaled. Scaling divides X coordinates by 2, and Y coordinates by 128/59. This maps the 1024 × 780 display into 512 by 360. This allows a program written for the 1024 × 780 terminal to run unchanged, and still display the entire picture (with some loss in resolution). The image doesn't cover the entire screen (only going to X = 512). The remainder can be used as a dialog area for alphanumeric text (see figure 4-20).

Unscaled mode shows a 720 by 360 subset of the 1024 × 780 picture. The area this covers can be changed by modifing the value of the relocatable origin (and redrawing the picture). The relocatable origin is subtracted from all incoming coordinates in unscaled mode. If this is set to 0,0 (the default) the range X = 0 to 719, Y = 0 to 359 will be displayed (see figure 4-21).

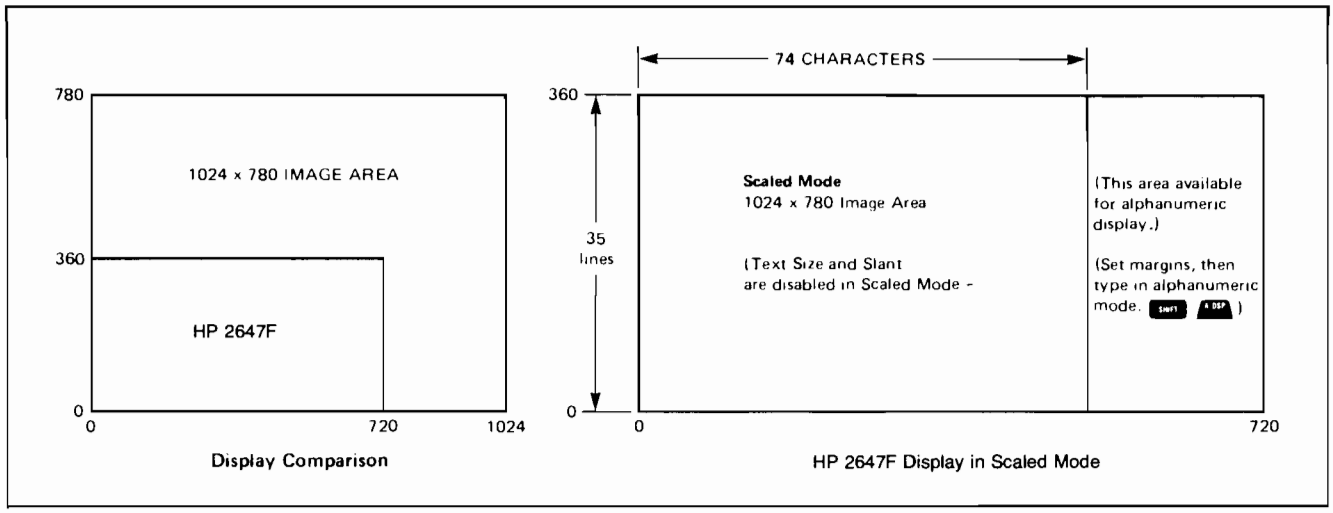
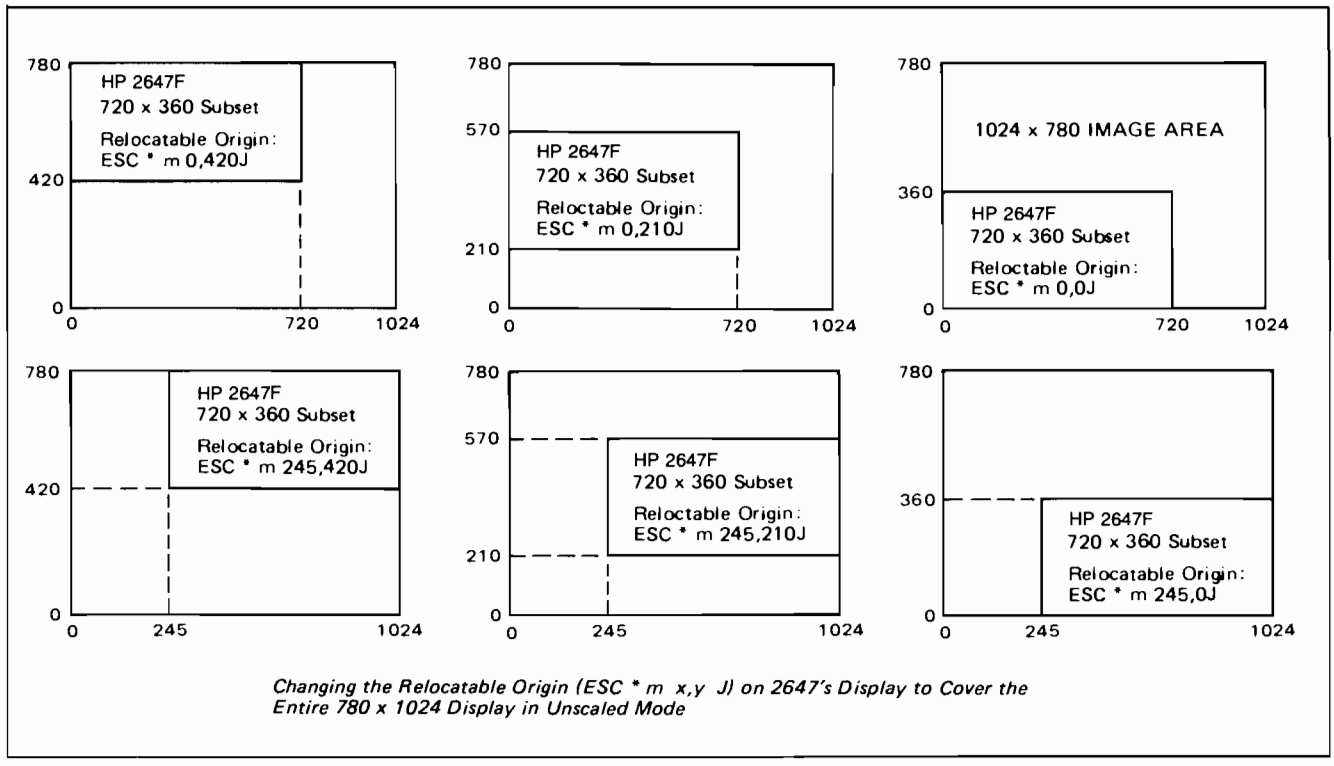


Figure 4-20. Scaled Data



*Changing the Relocatable Origin (ESC * m x,y J) on 2647's Display to Cover the Entire 780 x 1024 Display in Unscaled Mode*

Figure 4-21. Unscaled Data

Setting the origin to 0,360 would cover the area X = 0 to 719, Y = 360 to 719. To display an area larger than 720 × 360, you must change the scaling statements in the program. The advantage of unscaled mode over scaled mode is that unscaled allows you to use the entire available display area (see figure 4-21).

GRAPHICS DATA FORMAT. In Compatibility mode the graphics data is formatted as two-byte coordinate values. The lower five bits of each byte are used to make a 10 bit (0-1023) coordinate. Data sent to the terminal must have the "Y" coordinate sent first;
<Upper Y> <Lower Y> <Upper X> <Lower X>.

When data is returned to the computer (cursor position, etc.), the X coordinate is returned first; <Upper X> <Lower X> <Upper Y> <Lower Y>.

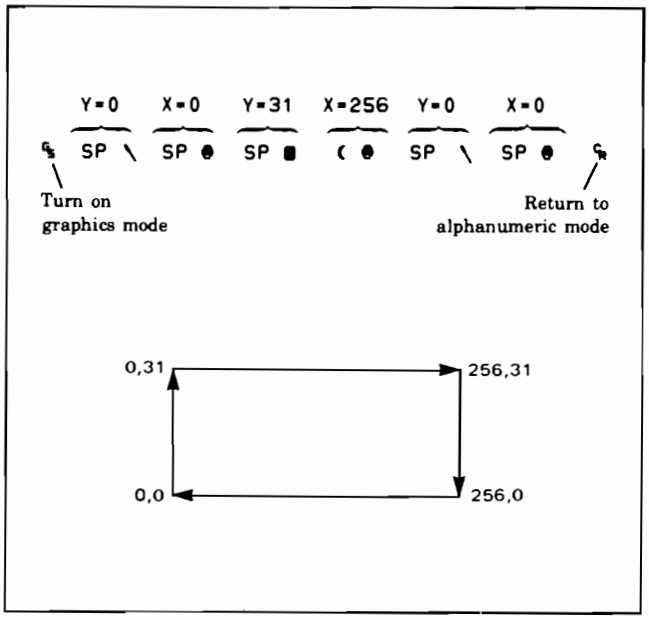
Data bytes sent to the terminal use bits 6 and 7 to indicate the byte is an Upper byte, a lower Y, or a lower X. Bit 8 (parity) is not used.

```
Bits
7 6
0 0  Upper X or Y byte
0 1  Lower X byte
1 1  Lower Y byte
```

These identifying bits allow you to send only the changed portion of a four byte address. The following data bytes must always be sent:

- Lower X byte
- Any changed byte
- Lower Y byte if the Upper X byte has changed

Table 4-15 can be used to determine address bytes. For example, to plot the points (0,0), (0,31), (256,31), (256,0) the following sequence would be used:



TEXT. Text can be placed in either the alphanumeric memory or in the graphics memory. If the terminal is set for alphanumeric text, the text will be sent to the alphanumeric memory. This is generally the most useful, as text can be scrolled, edited, erased, etc. without affecting the graphics image. If you select graphics text (ESC*ds), text will go into the graphics memory. Text to be written to the graphics memory can be scaled or rotated. (Refer to Graphics Text for additional information.)

When text is written to the graphics memory, the graphics cursor is moved to indicate where the next character will be stored. (The alphanumeric cursor is only used when data is stored in the alphanumeric memory.) This differs from terminals that have only one mode for text and display the graphics cursor only when waiting for graphic input from the user.

Scaled Mode Graphics Text. In Scaled Mode, text is initially written into the graphics memory, the size is fixed to allow for 35 lines of text. The text angle is set at 0 degrees and unslanted. The text origin is set to the left and bottom. These settings allow the "Page Full" feature to work properly and existing software to run without changes. If you do not require the Page Full feature, you can not change the text settings. You can redirect the text to the alphanumeric memory.

Unscaled Mode Graphics Text. In Unscaled Mode, the text size is unchanged and graphics text mode is not initially turned on. Text is stored in the alphanumeric memory unless the graphics text mode is specifically enabled.

Table 4-15. Coding of Compatibility Mode Graphics Data

X or Y Coordinate																Low Order Y		Low Order X	
																DEC.	ASCII	DEC.	ASCII
0	32	64	96	128	160	192	224	256	288	320	352	384	416	448	480	96	\	64	@
1	33	65	97	129	161	193	225	257	289	321	353	385	417	449	481	97	a	65	A
2	34	66	98	130	162	194	226	258	290	322	354	386	418	450	482	98	b	66	B
3	35	67	99	131	163	195	227	259	291	323	355	387	419	451	483	99	c	67	C
[illegible]																			
8	40	72	104	136	168	200	232	264	296	328	360	392	424	456	488	104	h	72	H
9	41	73	105	137	169	201	233	265	297	329	361	393	425	457	489	105	i	73	I
10	42	74	106	138	170	202	234	266	298	330	362	394	426	458	490	106	j	74	J
11	43	75	107	139	171	203	235	267	299	331	363	395	427	459	491	107	k	75	K
[illegible]																			
16	48	80	112	144	176	208	240	272	304	336	368	400	432	464	496	112	p	80	P
17	49	81	113	145	177	209	241	273	305	337	369	401	433	465	497	113	q	81	Q
18	50	82	114	146	178	210	242	274	306	338	370	402	434	466	498	114	r	82	R
19	51	83	115	147	179	211	243	275	307	339	371	403	435	467	499	115	s	83	S
[illegible]																			
24	56	88	120	152	184	216	248	280	312	344	376	408	440	472	504	120	x	88	X
25	57	89	121	153	185	217	249	281	313	345	377	409	441	473	505	121	y	89	Y
26	58	90	122	154	186	218	250	282	314	346	378	410	442	474	506	122	z	90	Z
27	59	91	123	155	187	219	251	283	315	347	379	411	443	475	507	123	{	91	[
[illegible]																			
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	← DEC.			
SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	← ASCII			

High Order X & Y

X or Y Coordinate																Low Order Y		Low Order X	
																DEC.	ASCII	DEC.	ASCII
512	544	576	608	640	672	704	736	768	800	832	864	896	928	960	992	96	\	64	@
513	545	577	609	641	673	705	737	769	801	833	865	897	929	961	993	97	a	65	A
514	546	578	610	642	674	706	738	770	802	834	866	898	930	962	954	98	b	66	B
515	547	579	611	643	675	707	739	771	803	835	867	899	931	963	995	99	c	67	C
[illegible]																			
520	552	584	616	648	680	712	744	776	808	840	872	904	936	968	1000	104	h	72	H
521	553	585	617	649	681	713	745	777	809	841	873	905	937	969	1001	105	i	73	I
522	554	586	618	650	682	714	746	778	810	842	874	906	938	970	1002	106	j	74	J
523	555	587	619	651	683	715	747	779	811	843	875	907	939	971	1003	107	k	75	K
[illegible]																			
528	560	592	624	656	688	720	752	784	816	848	880	912	944	976	1008	112	p	80	P
529	561	593	625	657	689	721	753	785	817	849	881	913	945	977	1009	113	q	81	Q
530	562	594	626	658	690	722	754	786	818	850	882	914	946	978	1010	114	r	82	R
531	563	595	627	659	691	723	755	787	819	851	883	915	947	979	1011	115	s	83	S
[illegible]																			
536	568	600	632	664	696	728	760	792	824	856	888	920	952	984	1016	120	x	88	X
537	569	601	633	665	697	729	761	793	825	857	889	921	953	985	1017	121	y	89	Y
538	570	602	634	666	698	730	762	794	826	858	890	922	954	986	1018	122	z	90	Z
539	571	603	635	667	699	731	763	795	827	859	891	923	955	987	1019	123	{	91	[
[illegible]																			
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	← DEC			
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?	← ASCII			

High Order X & Y

Example: 340Y,70X is found as follows:

340Y = 42 (upper Y) 116 (Lower Y) 70X = 34 (Upper X) 70 (Lower X)

340Y,70X → • t " F

COMMANDS

SECTION 5

INTRODUCTION

Control of mini-disc drives and HP-IB devices, such as printers and plotters, is effected using comands. Commands entered into the terminal locally from the keyboard or from a program running on a remote computer using escape sequences (see figure 5-1).

The terminal incorporates a feature called "cartridge tape emulation" to make it compatible with programs developed for 264X terminals containing tape units. Using this feature, data can be stored in disc files and manipulated as though on tape. As a result, some of the terminals commands, such as Rewind, which are obviously meant for tapes, can be executed, although the terminal contains no tapes. Tape emulator commands can originate from either the keyboard or a program.

This section is divided into four parts; the first part discusses data operations, the second part discusses commands entered locally from the keyboard, the third part concerns execution of a set of commands stored on the display or a disc file, and the fourth part discusses commands originating from a program running on a computer.

Table 5-1 lists the commands, gives a brief description of each command, and indicates the section in which the command is discussed in detail.

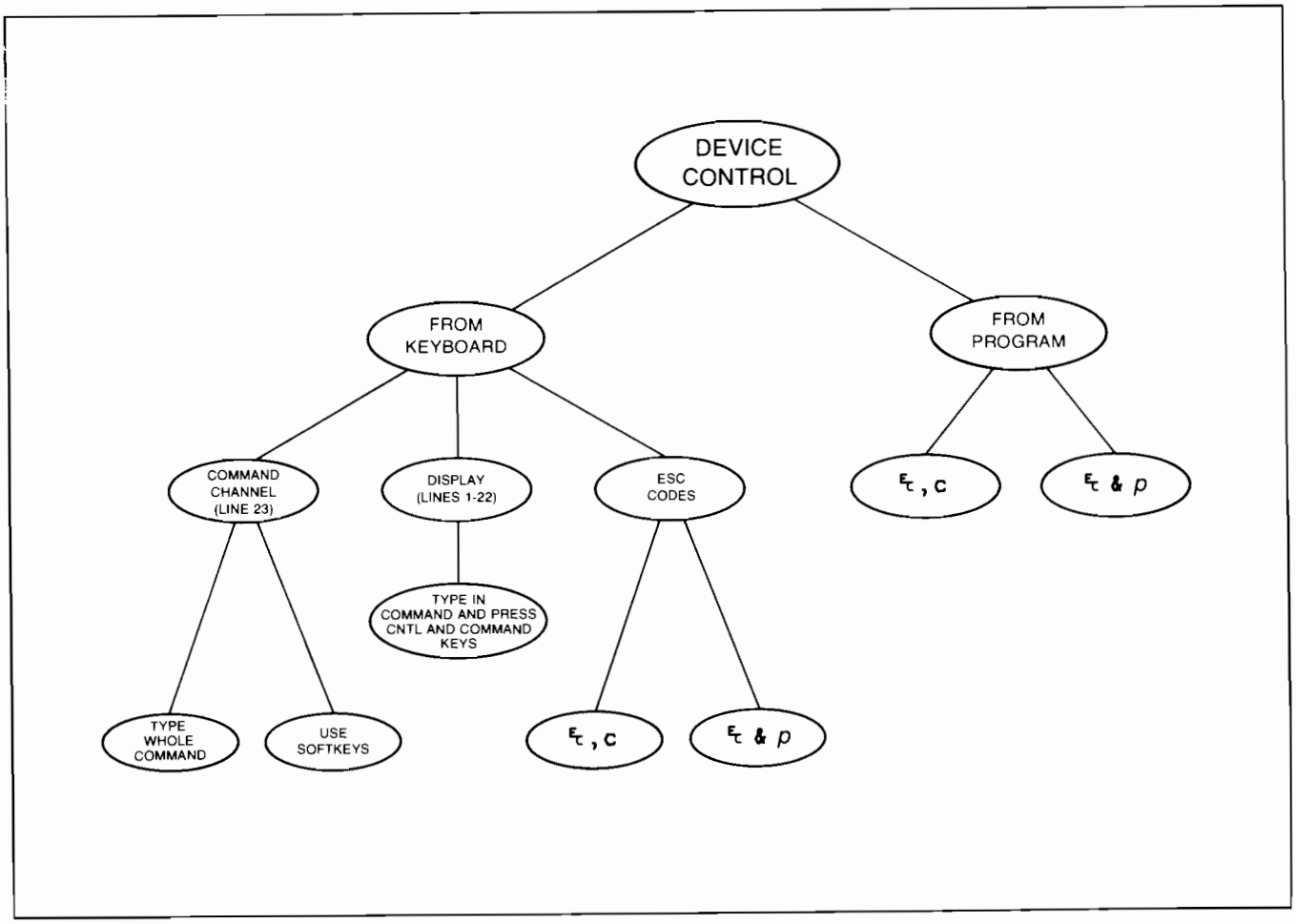


Figure 5-1. Means Available for Device Control

Table 5-1. Commands and Softkey Labels

COMMAND	PAGE	FUNCTION
APpend*	5-18	Add data to the end of an existing file.
Assign	5-19 6-4 7-2	Assign the device(s) or disc file(s) to the logical file names Source, Destination, LOG, Printer, CTUL, CTUR, or a filename selected by the user.
BYE	5-19 7-2	Ends use of current user.group label assigned by the HELLO command sequence to be used on the header and trailer pages on the shared printer listings.
CALENDAR	5-20	Displays a one-month calendar on the CRT with the specified date highlighted. If no date is specified, displays the month containing the current date (provided a date has been set using the Set command).
CLOse	5-20	Erases from the screen any data on the message, command,or user softkey labels line(s).
COmpare	5-20 6-15	Makes a byte-by-byte comparison of each record, file, or volume on the selected disc files or device(s).
Copy	5-21 6-9 7-3	Copies a selected amount of data (line, file, all data in the currently-selected workspace starting at the cursor line, or an entire volume) from the selected source (display, disc file, HP-IB device, or volume) to the selected destination (any destination device).
CReate*	5-22 6-3	Prepares either a disc or a file area on disc for data storage. For a disc (volume): a volume label is written on the disc and space is allocated for the directory. Any previously unformatted disc is formatted, using the default stagger before the label is written and directory space allocated. For a file: opens an area on disc for future storage of a file. The size and type (ASCII or binary) are selectable by the user.
DIAL	5-22 8-2	Operates through the 13265A Modem Pod to automatically dial a selected telephone number.
disc	5-4	The commands Append, Create, Format, Pack, Purge, Rename, and Unpurge, are accessed through the "disc" label.
DIsplay *	5-23	Enables entry and display of data on any of the four workspaces and the message and command lines.
EDIT	5-24 6-12	Enables transfer of data from a file to the display where it can be edited, then to another file as it rolls off the display screen.
Enable/ Disable	5-23	Turns on and off Edit, Record, and Verify modes and toggles the graphics/numeric keypad between operation as a graphics or numeric keypad.
EXecute	5-24	Executes a group of commands located on any selected source device. (Execution of the commands can be suspended before completion by the SUspend command and then resumed using the RESume command.)
EXIT	5-24	Terminates execution of commands being executed using the EXecute command. Also, terminates execution of a Terminal Basic program.
FIND	5-24	Positions a CTU Emulator file on disc to an absolute or relative (+ or −) file number. Also positions a CTU Emulator file on disc to the end-of-data mark.
FORmat *	5-25	Records, on a disc, the sector information required to make the disc compatible to the terminal file system. Enables selection of stagger. A disc must be formatted before it can be used.
forms		Accesses Forms (Format) mode which enables drawing forms for data entry, displaying the forms, and entry of data in the forms. (Refer to the Data Entry and Forms Design manual for information on Forms mode.)
HELLO	5-25 7-2	Enables selection of the names to appear as the user and group identities on the header and trailer pages of listings on shared printers.
Keypad	5-5	Enables selection of the graphics keypad on the keyboard for use either as a graphics or numeric keypad.
Mark	5-25	Closes a file when applied to a disc and writes a file mark on a CTU emulator file when applied to a "tape".
PAck *	5-25 6-14	Deletes all purged files from a disc and shifts the data on the disc to fill the vacated space.
PURGE *	5-26 6-13	Deletes a disc file or volume. (Refer to the UNpurge command.)
REName *	5-26 6-14	Enables renaming an existing disc file or disc (volume).
REPort *	5-26	Generated from a program to cause the terminal to send the completion status of the last command (successful or unsuccessful) to the program on the datacomm line.
RESume	5-26	Used to resume execution of a group of commands previously activated by the EXecute command and suspended by SUspend command.

Table 5-1. Commands and Softkey Labels (Continued)

COMMAND	PAGE	FUNCTION	COMMAND	PAGE	FUNCTION
REwind	5-27	Used to rewind the specified set of CTU emulator files to the beginning of the set of files.	Test	5-30 13-6	Enables testing the terminal, disc drive, the datacomm line, or the data path to a selected HP-IB device or terminal in a network.
SET	5-27	Enables setting the time (hours, minutes, and seconds) or date.	text		Accesses Text Preparation mode which enables editing and adjustment of text. (Refer to the Data Entry and Forms Design manual for information on Text Preparation mode.)
SHow	5-27 6-4	Depending on the accompanying parameter, displays the assignments recorded in the Assignment table, the name of the mounted disc (volume), the files on the currently- mounted disc, names of the purged files on the currently-mounted disc, the number of the CTU emulator file currently pointed to by the "file pointer", the time, or the date.	TRansfer	5-31 6-10	Copies binary (8-bit) data (line, file, or all data in the currently selected workspace starting with the cursor line) from any selected source device to any selected destination device. (The TRansfer command is similiar to the Copy command – refer to the "Data Movement" paragraph in this section.)
SKip	5-29	Positions a device to a relative line or beyond an end-of-data mark.	UNpurge *	5-31 6-14	Restores a previously purged disc file provided the PAck command has not been used on the disc (volume). Also, used to restore a previously-purged volume.
SUspend	5-30	Interrupts execution of a set of commands initiated using the EXecute command. Execution of the commands can be resumed by using the RESume commands.	VERify *	5-32 6-16	Checks the CRC's on the mounted disc (volume).
TELL	5-30	Enables sending a message of up to 80 characters to any destination device.			

NOTE: Commands followed by an asterisk (*) are accessed through the "disc" softkey.

CARTRIDGE TAPE EMULATION

Cartridge Tape Emulation is a feature which enables the terminal to respond to tape commands. The tape commands may originate from the keyboard or from escape sequences sent from a host computer. This feature is useful in data entry applications when Forms Mode data is stored on a diskette in local mode. Also it provides backward compatibility with other 264X terminals that use cartridge tape units (i.e., previously written programs to control the tape units can also be used to drive the Cartridge Tape Emulator). Refer to Section 6 for a discussion of tape emulation.

DATA OPERATIONS

The terminal operates on two types of data, referred to in this mannual as ASCII and binary. (The data type is significant in four commands; APpend, COmpare, Copy, and TRansfer.) The significant characteristics of the data types are described below:

ASCII: Normally, for ASCII-type data, only the seven low-order bits are considered significant. Examples of ASCII data include text, forms, data recorded from forms, and LISTing of a Basic program. In general, anything that can be represented on the terminal display can be represented as ASCII data.

BINARY: All eight bits of each byte are considered significant. Graphics raster data is an example of binary data.

Some devices are typed; that is, the terminal can tell whether their data is ASCII or binary. For example, each disc file is marked with a file type which tells the type of data it contains. The Graphics "device" is always considered to be binary.

NOTE: As used here, "device" can be an assignment device from the Assignment table or a "<device>" as listed on sheet 1, figure 5-3.

Other devices are untyped and can handle either type of data. Selected printers, DATACOMM, DISPLAY, and LEFT TAPE and RIGHT TAPE are untyped devices. (Although the devices LEFT TAPE and RIGHT TAPE are untyped, CTU emulator files are typed.)

DATA MOVEMENT

Two data movement modes correspond to the two data types; ASCII data movement mode and binary data movement mode. In ASCII mode, a byte of data with the high-order bit set may be translated into an escape sequence or ignored. In binary mode, no bytes are changed or discarded. Whenever possible, the terminal matches the

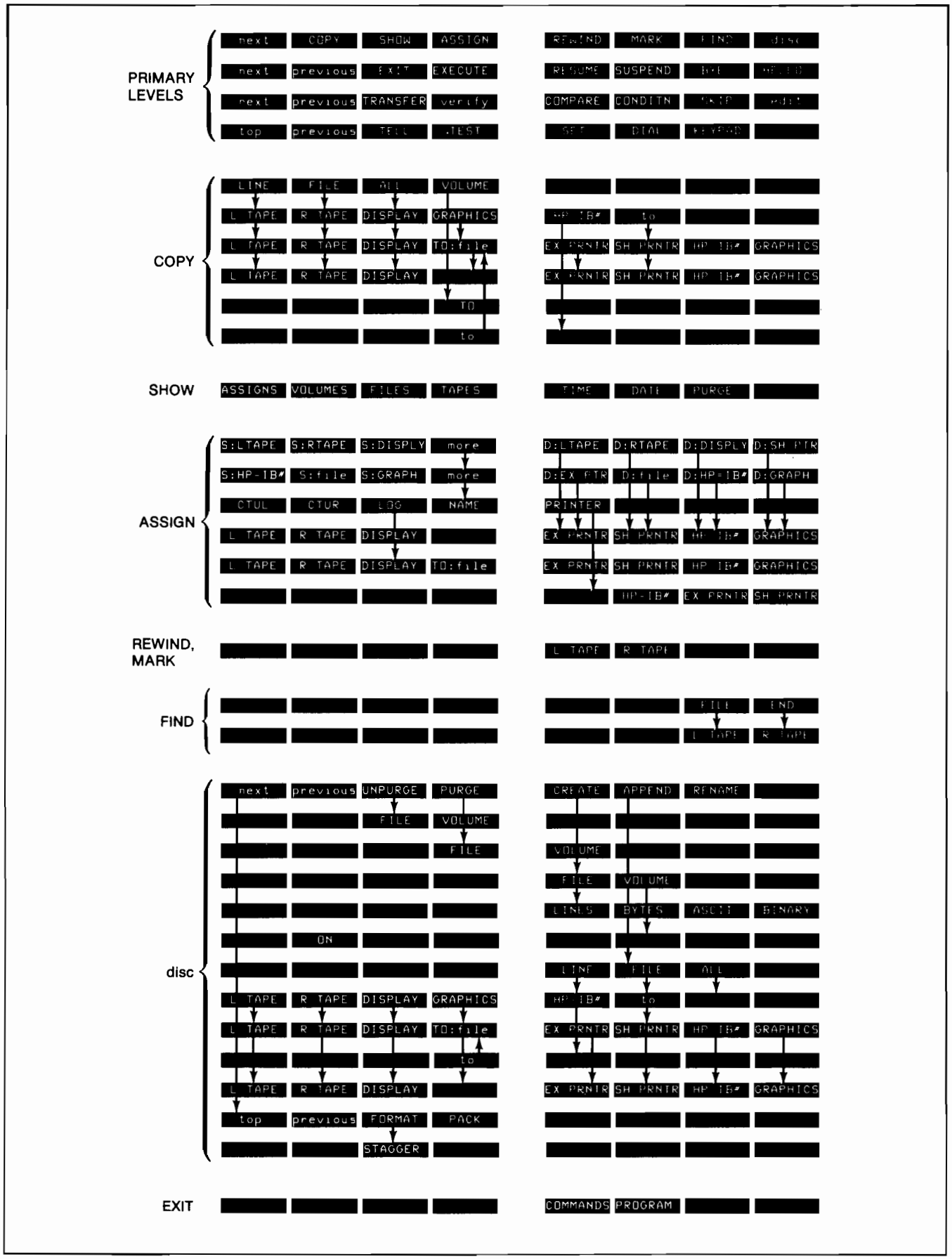


Figure 5-2. Command Tree

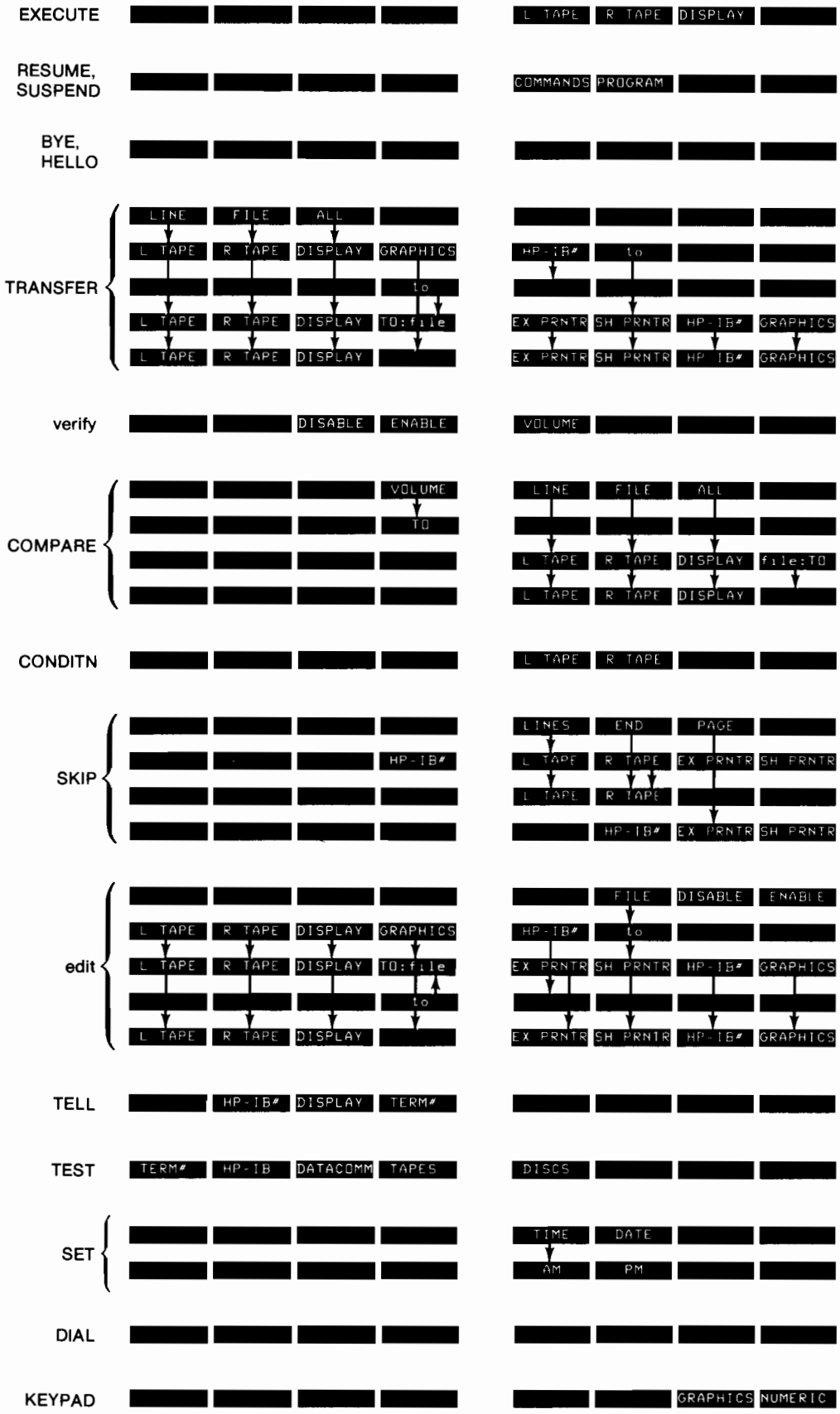


Figure 5-2. Command Tree (continued)

** <device> defined as
Left tape
Right tape
EXternal printer
SHared printer#<n> (where 0≤n≤7 i.e., SH#5)
TErminal#<p>
Hp-ib#<p>[#<s>[#<m>]]
where:
p = primary address
s = secondary address
m = module address
DAtacomm
Null

** <name> defined as
Source (only one may be selected)
Destination
LOG
PRINTER
user-assigned name (for disc files, cartridge tapes, printers, and Hp-ib devices)

** <volume name> defined as
user-assigned volume name (up to 6 alpha and/or numeric characters; an alpha character must be first)

** <disc file name> defined as
user-assigned disc file name (up to 10 alpha and/or numeric characters; an alpha character must be first). The volume name may be specified immediately following the disc file name, separated by a semicolon (;) (no spaces allowed). The disc drive unit number may be specified instead of the volume name. Use either Disc # <n> or D # <n> immediately following the disc file name separated by a colon (:) (no spaces allowed). *The volume name must be included when specifying file names on a volume mounted on disc drive no. 2.*

** <disc device> defined as
DISC#<n> where n is the drive unit no. (1 or 2)

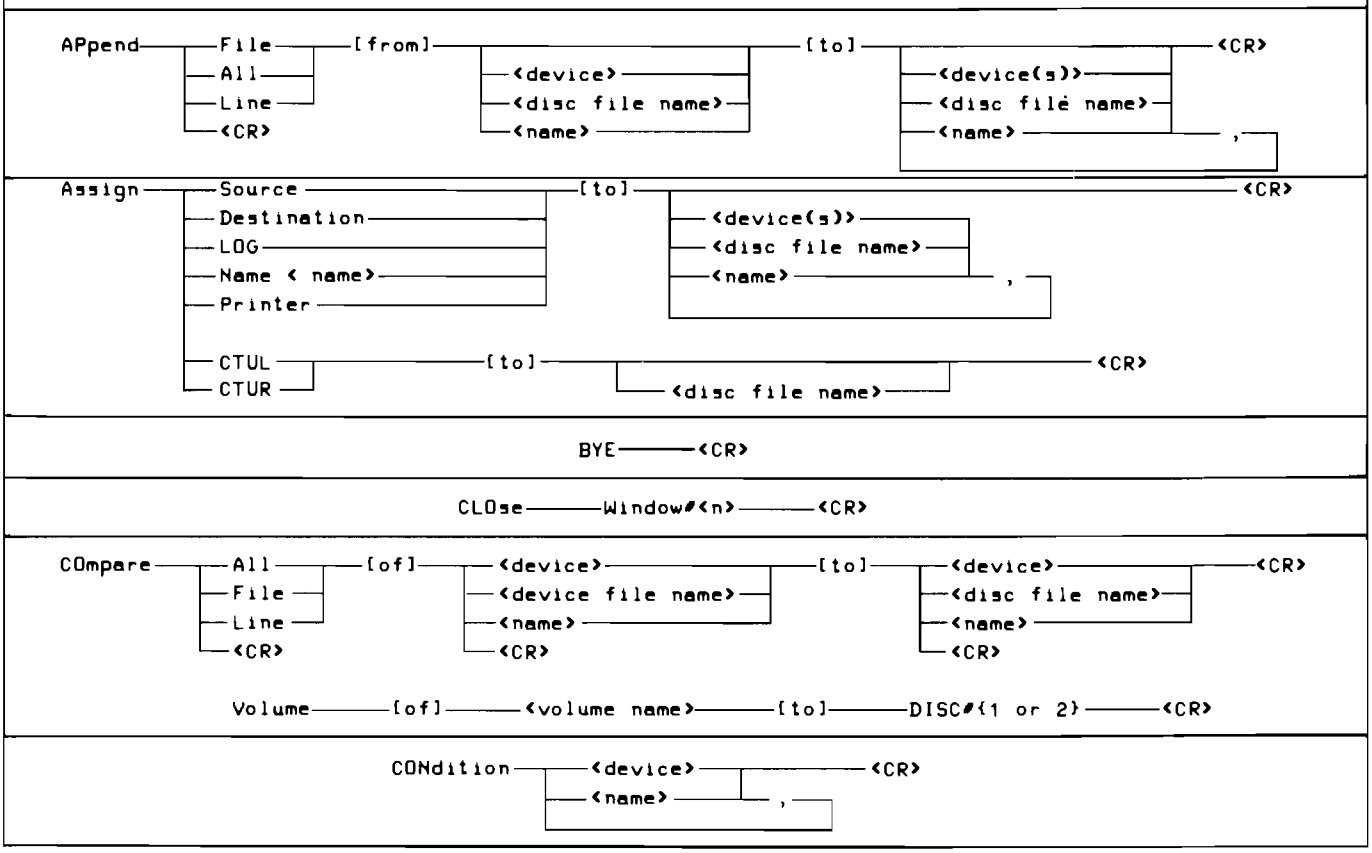


Figure 5-3. Command Syntax Diagram

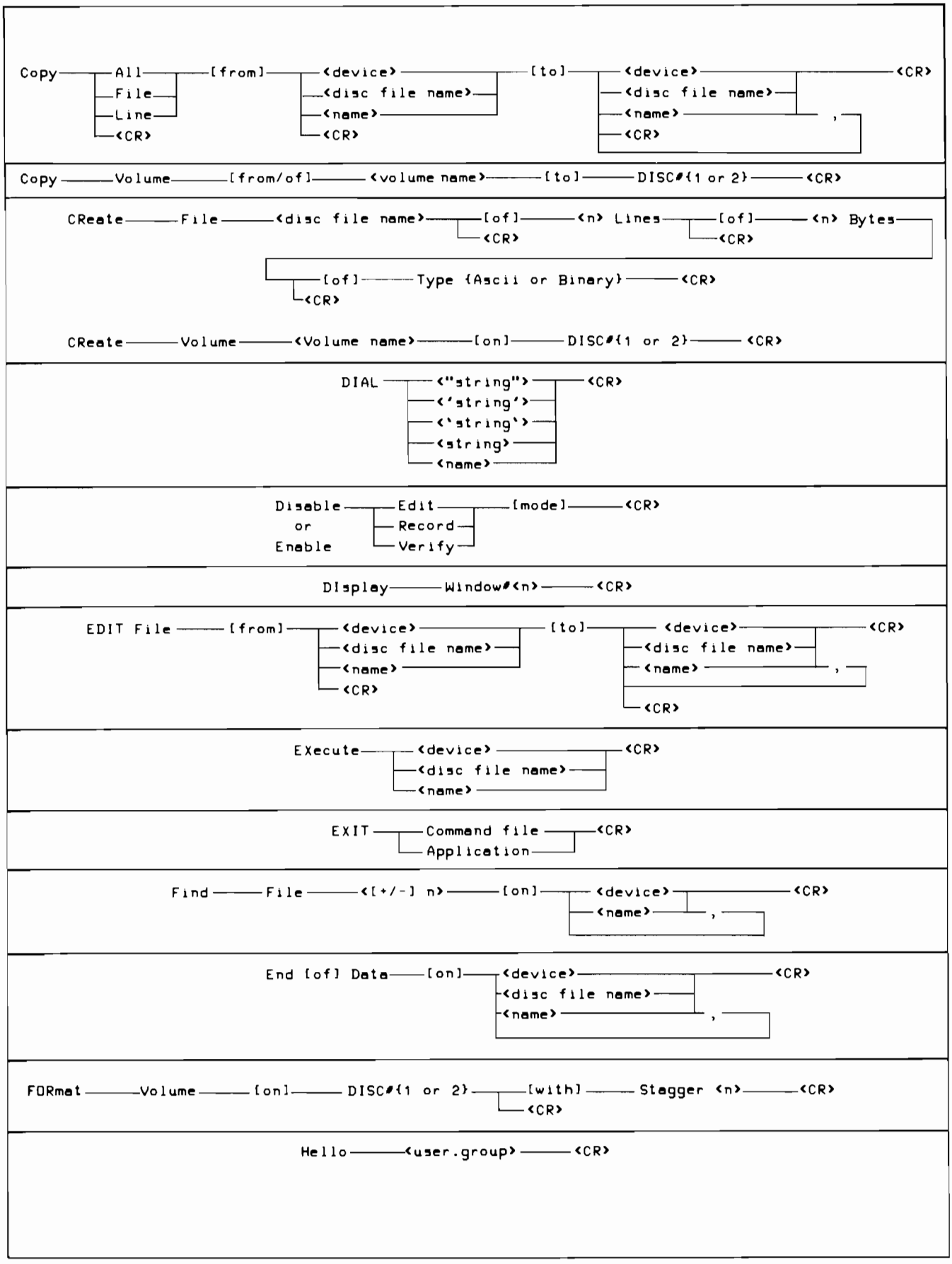


Figure 5-3. Command Syntax Diagram (continued)

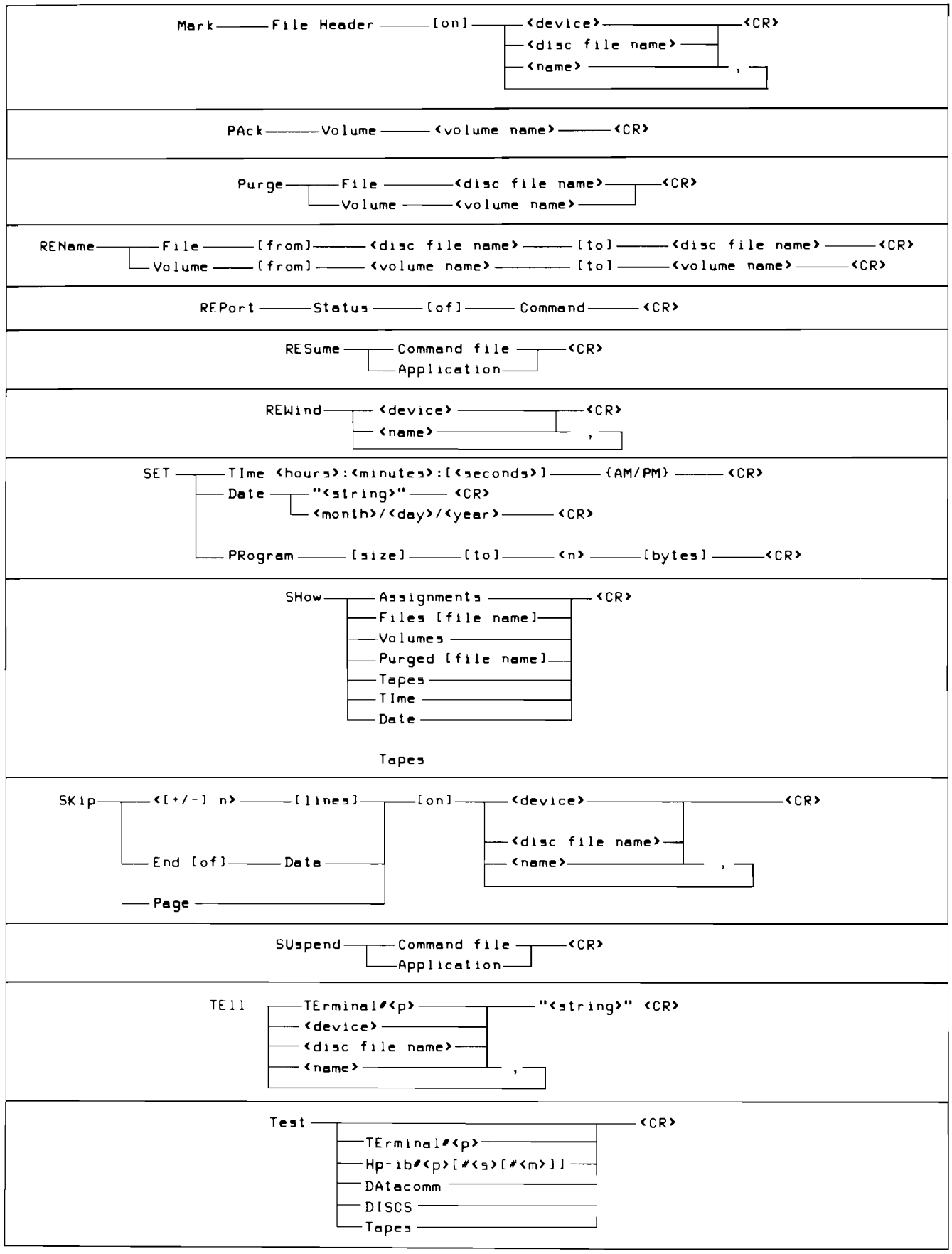


Figure 5-3. Command Syntax Diagram (continued)

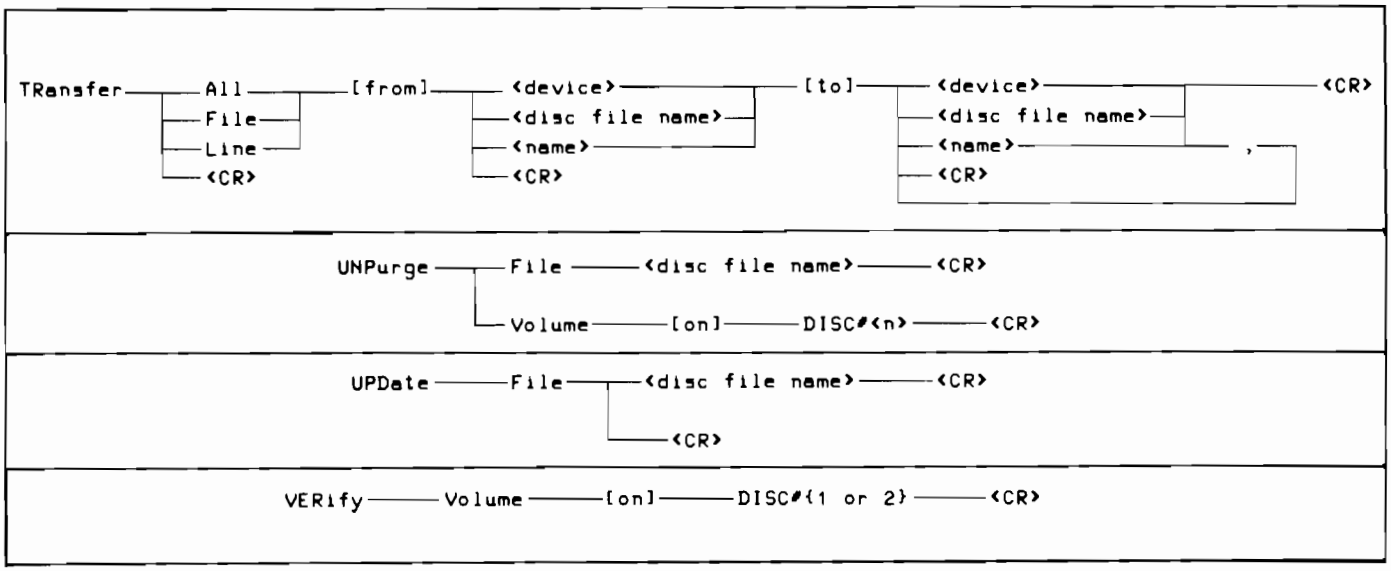


Figure 5-3. Command Syntax Diagram (continued)

data movement mode to the data being transferred. The data movement type for each of the four commands is selected as shown below.

COMMAND	"FROM" DEVICE TYPE	"TO" DEVICE TYPE	DATA MOVEMENT MODE
Copy (note 1)	ASCII	—	ASCII
	Binary	—	Binary
	Untyped	—	ASCII
TRansfer (note 1)	ASCII	—	ASCII
	Binary	—	Binary
	Untyped	—	Binary
APpend (note 2)	—	ASCII	ASCII
	—	Binary	Binary
	—	Untyped	ASCII
COmpare	ASCII	ASCII	ASCII
	Binary	Binary	Binary
	ASCII	Binary	(note 3)
	Binary	ASCII	(note 3)
	Untyped	ASCII	ASCII
	Untyped	Binary	Binary
	ASCII	Untyped	ASCII
	Binary	Untyped	Binary
	Untyped	Untyped	ASCII

NOTES:
1. If the "to" device is a disc file, the type (ASCII or binary) of the disc file is assigned to be the same as the data movement mode.
2. It is possible to APpend binary data to an ASCII file; but this is not recommended because the results are unpredictable.
3. If, for a COmpare command, both the "from" and "to" devices are typed and the types do not match, a "FILE TYPE CLASH" error message is displayed.

EXECUTING COMMANDS FROM THE KEYBOARD

The following paragraphs describe four methods of command entry from the keyboard, introduce the command tree, supply the command syntax, and discuss, in detail, keyboard entry of five commands (Assign, Set, Calendar, Show, and Tell).

Commands originating from the keyboard and concerned primarily with mini-disc drives are discussed in detail in Section 6, Disc Operations From the Keyboard, and commands concerned with printers are discussed in Section 7, Printer Operations From the Keyboard. Commands used to execute a set of commands stored on the display or on a disc file are discussed under the heading "Executing Commands from the Display or a Disc File". The remaining commands (Assign, Execute, Suspend, Resume, Exit, Set, Show, and Tell) are discussed under this paragraph heading.

Command Mode

Command mode enables easy entry of commands into the terminal. Alternate presses of the [COMMAND] key initiate and end Command mode.

COMMAND ENTRY. Four methods are available for entering commands into the terminal from the keyboard, two while in Command mode and two without the use of Command mode. Commands entered while in Command mode are entered on line 23 of the display.

Method 1: Using the Softkeys in Command Mode. In Command mode, commands may be entered using softkeys [f1] through [f8], located along the top of the keyboard. These keys are used in association with the softkey labels which are displayed on line 24 of the display in Command mode. For example, the third label from the left is associated with [f3], the third key from the left.

Each of the softkeys can be assigned multiple functions by changing the corresponding label on line 24. As shown in figure 5-2, many labels are available in the form of a "command tree". The top four rows of labels are the primary set of labels through which all remaining rows are accessed. The top row of labels is displayed after a power-up or hard reset and the other three rows of primary labels are accessed through this row, as shown by the arrows in the figure.

Except for the "next", "previous", and "top" labels, each label in the four primary rows accesses a set of secondary labels. These sets range in size from one to five rows.

When one of the softkeys is pressed in Command mode, a related portion of the command appears on line 23. This shows the user the completed portion of the command entry and serves as an aid to pressing the softkeys in the proper sequence. Figure 5-3 illustrates the complete commands and can be used as a guide to the next softkey to be pressed.

When the complete command is displayed in line 23, pressing the [RETURN] key initiates the command.

Example: Suppose that you wanted to copy all the data in the present display workspace to the minifloppy.

Step 1. Press [COMMAND] to enter Command mode.

Step 2. Press [f2], COPY to start the command sequence; the keys are now loaded to specify how much data is to be copied.

Step 3. Press [f3], ALL to specify that all data in the workspace, starting at the line containing the cursor, is to be copied (including data that has rolled off the bottom of the screen). The keys are now loaded with a list of "from" devices.

Step 4. Press [f3], DISPLAY to specify the display as the from device; the keys are now loaded with a list of "to" devices.

Step 5. Press [f4], TO:file and type "EXAMPLE" for a disc file name. (The "to" device selections remain loaded so that you may specify more than one "to" device.

Step 6. Press [RETURN] to indicate that the command sequence is complete. Execution of the command sequence begins when the [RETURN] key is pressed.

Step 7. Press [COMMAND] to remove the Command Line and soft key labels.

Method 2: Typing in the Command in Command Mode. While in Command mode, an alternative to using the softkeys is to type in the command on line 23 using the keyboard keys. The command typed in should conform to the command syntax as shown in figure 5-3, including the words enclosed in brackets (such as "[from]" and "[to]" in the Copy command). The [RETURN] key must be pressed to initiate the completed command.

Method 3: Typing in the Command While Not in Command Mode. A command can be typed into any of lines 1 through 24 and initiated by placing the cursor on the line containing the command and pressing, simultaneously, the [CNTL] and [COMMAND] keys. As in method 2, the command must be as shown in figure 5-3.

Method 4: Typing in the Escape Sequence. While not in Command mode, a command may be entered by typing the corresponding escape sequence on any of lines 1 through 24. The command is initiated by pressing the [RETURN] key. Refer to "Executing Commands From a Program" later in this section for the escape sequences. Either the ESC,c or ESC&p type sequence can be used.

CORRECTING COMMAND SEQUENCES. If you press the wrong key or leave out part of the command while typing the command sequence, an error message will appear when the [RETURN] key is pressed. At this point, you can correct the command line using the edit keys [INSERT CHARACTER] and [DELETE CHARACTER], or start over by using the [DELETE LINE] key. After the correction has been made, press the [RETURN] key to start execution.

Example: Suppose that a diskette is mounted in the drive and that you attempted to copy a disc file named "EXAMPLE" to the display but the file did not exist on the diskette; an error message would be printed out on the display.

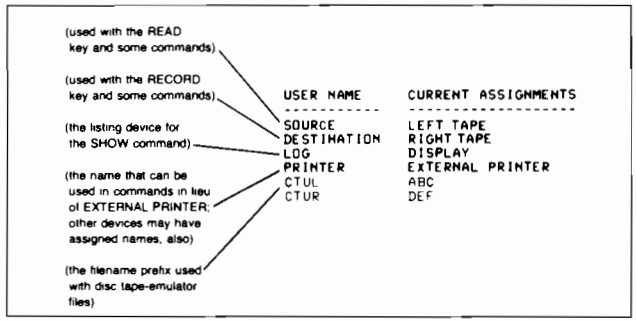
Step 1. The command line may be cleared by using either [DELETE LINE] or [HOME UP], [CLEAR DISPLAY].

Step 2. Re-enter the command and specify the correct file name. (If you are unsure of the filename, press SHOW, FILES to list the active files on the diskette.)

Assign Command

The Assign command is used to alter the entries in the Assign table (i.e., "Source", "Destination", "Log", "Printer", "CTUL", "CTUR", and the name of any of the devices). The Show Assignments command lists the current assignments on the "log" device (default is the display). A sample listing is shown below.

When the terminal is turned on or when the [RESET TERMINAL] key is pressed twice within 1/2 second, the Assign table is set to the defaults shown below.



THE "SOURCE" DEVICE. The "source" device is used primarily by the [READ] key. When the [READ] key is pressed, the "source" device is read to the display. (If the terminal is in Remote mode ([REMOTE] key down), the data is read from the "source" device to datacomm.)

The default source "left tape" may be changed by using the Assign command.

THE "DESTINATION" DEVICE. The "destination" device is used primarily by the [RECORD] key. When the [RECORD] key is pressed, the display is recorded on the "destination" device. (If the terminal is in Remote mode, the data is recorded from datacomm to the "destination" device.)

The default destination "right tape" may be changed using the Assign command. (More than one "destination" device may be specified.)

THE "LOG" DEVICE. The assigned "log" device lists the data resulting from any of the Show commands. Any of the devices may be assigned to the "log" device (i.e., display, a tape emulator file, or any of the printers).

A disc file can be assigned as a log device; in which case, the data will be written on the file.If a file with the given name doesn't exist, one is created and written on. If one exists and already contains data, the Show command data is appended to it.

More than one device can be assigned as the "log" device, in which case, the output is listed on each one. A device assigned as a "log" device can also be assigned to other entries on the Assign table (e.g., the left tape could be assigned as the "destination" device and the "log" device).

THE "PRINTER". The "printer" entry in the Assign table shows which printer (external printer, shared printer, or HP-IB printer) is equated to the name "printer" This printer will respond to the ESC&p4D escape sequences from a host computer. That is, a program which contains an ESC&p4D sequence which is directed to the terminal will assign the device equated to the "printer" in the Assign table as the destination. (More information on ESC&p escape sequences appears later in this section.

The default "printer" assignment is the external printer.

The "printer" entry in the Assign table allows specifying the name "printer" instead of pressing [EX PRNTR], [SH PRNTR], or [HP-IB#] when entering a command sequence.

The shared printer or the HP-IB printer may be equated to the name "printer" by executing the Assign Printer command. As an example, if your terminal is connected to a shared printer (the Show Volumes command will list the shared printers if they are cabled to the terminal, powered-on, and on line), the name "printer" could be equated to the shared printer by the Assign Printer command.

"CTUL" AND "CTUR" ASSIGNMENTS. The "CTUL" (left tape) and "CTUR" (right tape) entries in the Assign table enable assignment of CTU emulator status to a group of disc files (consisting of from one to 255 files). This enables manipulation of each file in the group using tape commands.

To assign CTU emulator status to a group of disc files, a group name is assigned to either CTUL or CTUR in the Assign table using the Assign command. Each file in the group must be named with this name followed by a number indicating its sequential location in the group on disc. For example, if a group of five files are to be assigned CTU emulator status and the name "Murray" is assigned to "CTUL" as the group name in the Assign table, the files in the group must be named Murray1, Murray2, . . . Murray5. With CTU emulator status thus assigned to these files, they are susceptible to all commands addressing the left tape. Refer to Section 6 for further information on CTU emulation.)

Any name up to 10 characters (alpha and/or numeric; an alpha character must be first) may be assigned to any device (display, graphics memory, disc file, any printer, or any HP-IB device) through the Assign Name command. The assigned name may be used in the command sequence instead of the normal device name; although, the normal device name may be used at any time.

This allows meaningful names to be assigned for special situations (e.g., "BILLING" could be assigned to a disc file, "PACKINGSLIP" could be assigned to a printer). Then, the commands would use these names (e.g., COPY FILE FROM BILLING TO DISPLAY, or COPY ALL FROM DISPLAY TO PACKINGSLIP).

Example: Assign "PACKINGSLIP" to the external printer.

Step 1. Press [COMMAND], [ASSIGN], [more], [more], [NAME], PACKINGSLIP, [to], [EXPRNTR], [RETURN].

Step 2. Show the change in assignment by pressing [SHOW], [ASSIGN], [RETURN].

```
USER NAME     CURRENT ASSIGNMENTS
-----------   -------------------
SOURCE        EDSMEMO
DESTINATION   TOMS4CAST
LOG           EXTERNAL PRINTER
PRINTER       SHARED PRINTER#6
CTUL          ABC
CTUR          DEF
PACKINGSLIP   EXTERNAL PRINTER
```

Step 3. Exit command mode by pressing [COMMAND] again.

Section 6 contains additional examples of use of the Assign command as used for disc operations.

Set Command

The Set command is used to set the terminal's internal clock, the date, and to set an alarm.

SET TIME

The Set Time command sets the terminal s internal clock. The time is entered in <hours>:<minutes>:<seconds> AM or PM format. (Seconds are optional.) A semicolon (:) separates the values. Once set, the time will not require resetting until the terminal is turned off.

The time may be displayed by the Show Time command described later. Also, terminal time is used by the Alarm function.

Example: Enter the time 10 hours, 17 minutes, 45 seconds, AM.

Step 1. Press [COMMAND], next, next, next, SET, time, 10:17:45, AM, [RETURN].

Step 2. Show the terminal time on the display. Press SHOW, TIME, [RETURN].

```
10:18:10 AM
```

Step 3. Exit command mode by pressing [COMMAND] again.

SET DATE

The Set Date command allows you to enter the date in a string format of up to 30 characters, or in a `mm/dd/yyyy` format, where `mm` is the month (1 to 12), `dd` is the day (1 to 31), and `yyyy` is the year (1929 to 2018).

NOTE: Once set, the date stored cannot be eliminated except by a hard reset or turning the terminal off. Also, it cannot be changed except by using the SET DATE command with the format:

```
mm/dd/yyyy
```

An attempt to change the date by the string method has no effect.

Setting by the String Method. The string must be enclosed in quotes. Once set by the string method, the date will not change. However, it will be destroyed by a hard reset. The date set by this method is printed on the shared printer listings only.

Example: Set the date to "Thursday, September 25, 1981".

Step 1. Press [COMMAND], next, next, SET, DATE, "Thursday, September 25, 1981", [RETURN].

Step 2. Show the date on the display. Press SHOW, DATE, [RETURN].

```
Thursday, September 25, 1981
```

Step 3. Exit command mode by pressing [COMMAND] again.

Setting by the mm/dd/yyyy Method. Once set, the date will not require resetting unless the terminal is turned off. The days are controlled by the internal clock (i.e., when the clock reaches 12:00 PM, the day is incremented). The year can also be specified with two digits (`yy`); in which case, the two missing first digits are assumed to be "19". The date, as set by this method, is used as the "create date" for disc files, and it is printed in shared printer listings.

One, two, or three numbers can be entered, separated by slashes (/), dashes (-), periods (.), or commas (,). If only one is entered, it is taken to be the date, and the previously-set month and year are used. If two numbers are entered, they are taken to be the month and date and the previously-set year is used. If an invalid date is entered, an error message is generated and the previously-set date is used. There is no default date provided for power-on time and any files or volumes created before the date is set will not have a date/time stamp. The first Set Date command after a power-on must have all three numbers provided since there are no previously-set values to use for the default values. Once set, the date is not changed by either a soft or hard reset.

Example: Set the date to July 23, 1980.

Step 1. Press [COMMAND], next, next, next, SET, DATE, 7/23/80, [RETURN].

Step 2. Show the date on the display. Press SHOW, DATE, [RETURN].

```
WED, JUL 23, 1980
```

Step 3. Exit command mode by pressing [COMMAND] again.

SET ALARM

The Alarm commands allow you to specify a time that an alarm, message, and/or command will alert you or cause the terminal to perform a given function. The commands are as follows:

COMMAND	FUNCTION
ALARM ON/OFF	Turns the alarm function on and off.
ALARM RING/SILENT	Causes the terminal's bell to ring continuously when the alarm time is is reached. ALARM SILENT command turns off the bell.
ALARM TIME	Specifies the alarm time. The terminal's internal clock is the reference. (See Set Time command.)
ALARM MESSAGE	Allows you to specify a message (up to 31 characters) to be displayed in the message line of the display when alarm time is reached.
ALARM COMMAND	Allows you to specify a command (up to 31 characters) to be executed when alarm time is reached.
ALARM SHOW	Displays the status of ALARM ON/OFF and ALARM RING/SILENT.
ALARM	Displays the current alarm time, message, and command stored in the terminal.

Example: Set the terminal to the current time, then set the time to 2 minutes later with alarm ring on. Display the message "DENTIST APPOINTMENT."

Step 1. Set the terminal time to 2:00 PM. Press [COMMAND], [next], [next], [next], [SET], [TIME], 2:00 PM, [RETURN].

Step 2. Enter `ALARM RING` into the command line.

Step 3. Enter `ALARM MESSAGE Dentist Appointment` into the command line.

Step 4. Enter `ALARM TIME 2:02 PM` into the command line.

Step 5. Enter `ALARM ON` into the command line.

Step 6. When the alarm rings, you may turn it off by entering `ALARM SILENT` or `ALARM OFF` into the command line.

Example: Set the alarm time and cause the terminal to execute a Show Assignments command.

Step 1. Press [COMMAND] to display the command line.

Step 2. Enter `ALARM COMMAND SHOW ASSIGNMENTS` in the command line.

Step 3. Enter `ALARM TIME 2:30 PM` into the command line.

Step 4. Enter `ALARM ON` into the command line.

Step 5. The Show Assignments command will display the assign table on the display at 2:30 PM.

```
USER NAME     CURRENT ASSIGNMENTS
----------    -------------------
SOURCE        LEFT TAPE
DESTINATION   RIGHT TAPE
LOG           DISPLAY
PRINTER       EXTERNAL PRINTER
CTUL          CTUL???
CTUR          CTUR???
```

Example: Set the alarm function to log-on the HP 3000 Computer System, at 12:00 midnight, call the Editor, read a disc file into the editor, print it, and log-off.

Step 1. Clear the display, and type the following text to be sent to the Editor:

```
This is the first line.
This is the second line.
This is the third line.
This is the fourth line.
//
```

Step 2. Copy the text to a disc file file named "TEXTSTUFF".

Press [HOME UP] to home the cursor.

Press [COMMAND], [COPY], [FILE], [DISPLAY], [TO:file], TEXTSTUFF, [RETURN].

Press [COMMAND] again to return to normal mode.

Step 3. Type the following onto the display (A blank line first for the carriage return to cause the colon prompt):

```
HELLO USER.ACCOUNT
EDITOR
ADD
```

Step 4. Copy it to a disc file named "LOGON".

Press [HOME UP] to home the cursor.

Press [COMMAND], [COPY], [FILE], [DISPLAY], [TO:file], LOGON, [RETURN].

Step 5. Clear the display, and type the following onto the display.

```
LIST ALL, OFFLINE
EXIT
YES
BYE
```

Step 6. Copy it to a disc file named "LOGOFF".

Press [HOME UP] to home the cursor.

Press [COMMAND], [COPY], [FILE], [DISPLAY], [TO:file], LOGOFF, [RETURN].

Press [COMMAND] again to return to normal mode.

Step 7. Create an execute file called "DOIT". Type the following on the display.

```
COPY FILE LOGON TO DATACOMM
COPY FILE TEXTSTUFF TO DATACOMM
COPY FILE LOGOFF TO DATACOMM
ALARM OFF
```

Step 8. Copy it to a disc file named "DOIT". Press HOME UP to home cursor.

Press COMMAND, COPY, FILE, DISPLAY, TO:file, DOIT, RETURN.

Step 9. Enter ALARM COMMAND DOIT into the command line.

Step 10. Enter ALARM TIME 12:00 PM into the command line.

Step 11. Enter ALARM ON into the command line.

Step 12. Press COMMAND to exit command mode.

Step 13. Press down the REMOTE key.

CALENDAR

The CALENDAR command is used to display a one-month calendar containing the specified date with the date highlighted. If no date is specified, the current date is used (the date previously set with the SET command).

Example: Display a calendar containing the current date. If no date has been entered previously, ennter the current date using the SET command.)

Step 1. Press COMMAND, type in CALENDAR, then press RETURN.

SHow

The Show command lists the assignment table, volume name(s) of mounted diskettes, shared printers that are available and HP-IB devices that are in use, active or purged disc files on the mounted volume (diskette), and present file number, terminal time, and terminal date. All listings are output to the current LOG device (display is the default).

SHOW ASSIGNMENTS. The Show Assignments command lists the Assign table. Refer to the Assign command, described previously, for an explanation of the entries.

Example: List the Assign table.

Step 1. Press COMMAND, SHOW, ASSIGNS, RETURN.

```
USER NAME     CURRENT ASSIGNMENTS
-----------   -------------------
SOURCE        DISPLAY
DESTINATION   HP-IB#3
LOG           DISPLAY
PRINTER       EXTERNAL PRINTER#6
CTUL          CTUL???
CTUR          CTUR???
```

Step 2. Exit command mode by pressing COMMAND again.

SHOW VOLUMES. The Show Volumes command lists the volume names of diskettes mounted in the mini disc drives, the shared printer numbers that are on-line to the terminal, and the addresses of any HP-IB devices that are in use.

Example: List the volumes.

Step 1. Press COMMAND, SHOW, VOLUMES, RETURN.

Step 2. Exit command mode by pressing COMMAND again.

NOTE: An explanation of the disc portion of the listing is contained in Section 6.

```
VOLUME NAME              HP-IB ADDRESS

Shared Printer#          6
HP-IB Device             3

Local   Disc      Start of    Space  Sector   Create       Create   Write
Volume  Type#Unit  Free Space wLeft  Stagger  Date(M/D/Y)  Time     Protect
------  --------- ---------- -----  -------  -----------  -------  -------
SALES   mfD#1        860       196     9      8/12/81      8:05 AM   NO
SHIPMT  mfD#2        750       306     9      8/14/81      2:54 PM   NO
```

SHOW FILES. The Show Files command lists the active files on either the integral disc (disc 1), the optional external disc (disc 2), or both. If the volume or device is specified in the command, only the files on the specified volume or disc are listed. If @ is specified, the files on disc 1 are listed, provided the disc is accessible. If it is not accessible, the files on disc 2 are listed. The simple Show Files command, not followed by any specification, lists the files on both discs.

Example: List the active files on both discs.

Step 1. Press [COMMAND], **SHOW**, **FILES**, [RETURN]

```
Volume • SALES

File        File     Start     Size       Create          Create
Name        Type     Address   (sectors)  Date (M/D/Y)    Time
--------    -----    -------   ---------  ------------    --------
FEBSALES    Ascii      32        49       3/21/81          4:13 PM
MARSALES    Ascii      81        19       4/18/81         11:32 AM
APRSALES    Ascii     100        18       5/16/81          9:21 AM
MAYSALES    Ascii     118         5       6/20/81          1:34 PM

Volume • SHIPMT

File        File     Start     Size       Create          Create
Name        Type     Address   (sectors)  Date (M/D/Y)    Time
---------   -----    -------   ---------  ------------    --------
FEBSHIPMT   Ascii      32        12       3/7/81          12:10 PM
MARSHIPMT   Ascii      44        14       3/4/81           7:38 AM
APRSHIPMT   Ascii     126        39       5/7/81           8:47 AM
```

Step 2. Exit command mode by pressing [COMMAND] again.

SHOW TAPES. The Show Tapes command indicates the present file position for each tape.

Example: Show the file position of both CTUL and CTUR.

Step 1. Press [COMMAND], **SHOW**, **TAPES**, [RETURN].

```
DEVICE NAME   FILE NUMBER   INCHES REMAINING
-----------   -----------   ----------------
LEFT TAPE     0             0
RIGHT TAPE    2             0
```

Step 2. Exit command mode by pressing [COMMAND] again.

SHOW TIME. The Show Time command lists the current terminal time. The time is set by the Set Time command discussed previously.

Example: Show the current terminal time.

Step 1. Press [COMMAND], **SHOW**, **TIME**, [RETURN].

```
5:09:15 PM
```

Step 2. Exit command mode by pressing [COMMAND] again.

SHOW DATE. The Show Date command lists the current terminal date. The date is set by the Set Date command discussed previously.

Example: Show the terminal's current date.

Step 1. Press [COMMAND], **SHOW**, **DATE**, [RETURN].

```
FRI, SEP 5, 1980
```

Step 2. Exit command mode by pressing [COMMAND] again.

SHOW PURGED. The Show Purged command lists the purged files on the mounted diskette. (Purged files may be recovered to active status if the diskette has not been packed to remove them.)

Example: Show the purged files on the mounted diskette.

Step 1. Press [COMMAND], **SHOW**, **PURGE**, [RETURN].

```
Volume • INVTRY

File        File     Start     Size       Create          Create
Name        Type     Address   (sectors)  Date (M/D/Y)    Time
-------     ------   -------   ---------  ------------    --------
HP2642A     Purged     58        29       8/4/80           9:21 AM
HP264SA     Purged     87        39       8/4/80           9:55 AM
HP2648A     Purged    165         2       8/4/80          11:06 AM
```

Step 2. Exit command mode by pressing [COMMAND] again.

TELL

The Tell command allows you to send messages to any device. This is particularly useful when operator intervention is required while executing command files. The message may be up to 80 characters and must be enclosed in quotes. No file mark is written at the end of the message.

Abbreviated Tell command sequences allow more room on the command line for the message. The abbreviated command sequences are given in the Appendix at the end of the manual.

The command softkeys allow you to select HP-IB#, DISPLAY, and/or TERM# as destinations. You may enter a disc file name, left or right tape, and/or one or more printers. Multiple dstinations are permitted.

The TERM# destination allows you to send messages to other terminals that are connected in a shared peripheral network.

Example: Tell the display "When ready to continue, type RESUME into the command line".

Step 1. Press [COMMAND], **next**, **next**, **next**, **TELL**, **DISPLAY**, "When ready to continue, type RESUME into the command line" [RETURN].

Step 2. Exit command mode by pressing [COMMAND] again.

Example: Tell the external printer "The following data is from Ken's marketing survey."

Step 1. Press [COMMAND], **next**, **next**, **next**, **TELL**, EXTERNAL PRINTER "The following data is from Ken's marketing survey", [RETURN].

Step 2. Exit command mode by pressing [COMMAND] again.

Example: Tell terminal#29 "Print your data on shared printer#6".

Step 1. Press [COMMAND], [next], [next], [next], [TELL], [TERM#], 29, "Print your data on shared printer#6", [RETURN].

Step 2. Exit command mode by pressing [COMMAND] again.

EXECUTING COMMANDS FROM THE DISPLAY OR A DISC FILE

The Execute command allows command sequences to be executed from a display workspace or a disc file). Each command sequence must be on a separate line or record. If user intervention is required, at any point in the execution (such as inserting a diskette or a tape), the Suspend command sequence may be used.

The Resume command sequence returns control to the device containing the command file. The Exit command terminates execution of the command sequences from the device.

Pressing [CNTL] [SOFTKEYS] displays the command labels for the [f1] through [f8] softkeys without the command line. When the appropriate softkey are pressed to build a command sequence, the command words are entered on the display at the current cursor position. (The command sequence is not executed.)

To get the top level of the command tree back into the label line, press [CNTL] [SOFTKEYS] again. This function is particularly useful when constructing Execute Files.

Example: Enter the SHOW ASSIGNMENTS command sequence into the present display workspace.

Step 1. Press [CNTL] [SOFTKEYS] ; the label line will show the command labels.

Step 2. Press [SHOW], [ASSIGN].

The commands are entered at the current cursor position on the display. [RETURN] terminates the sequence.

Example: Execute a list of command sequences stored on the diskette. The command sequences do the following:

- Read files 1 through 4 on the disc to display workspaces 1 through 4.
- Pause for user editing of the displayed data.
- Copy the edited data in each display workspace to files 1 through 4 on the disc.

Step 1. Insert a formatted and named diskette into the mini disc drive; be sure that the diskette is not protected. (Refer to Section 6 for formatting, naming and write-protecting diskettes.)

Step 2. Type a few lines on the display, then position the cursor to the first line. Copy the lines on the display to the disc file named "FILE1". Press [COMMAND], [COPY], [FILE], [DISPLAY], [TO:file], FILE1, [RETURN]. Press [COMMAND] to exit command mode.

Step 3. Repeat step 2 three more times using disc file names "FILE2", "FILE3", and "FILE4". (This provides the four files of data on the diskette).

Step 4. Clear the display, then type the command sequences listed below.

```
DISPLAY WINDOW#1
COPY FILE FROM FILE1 TO DISPLAY
DISPLAY WINDOW#2
COPY FILE FROM FILE2 TO DISPLAY
DISPLAY WINDOW#3
COPY FILE FROM FILE3 TO DISPLAY
DISPLAY WINDOW#4
COPY FILE FROM FILE4 TO DISPLAY
DISPLAY WINDOW#4
TELL DISPLAY "PRESS 'CNTL ' 'PAGE NEXT' TO EXAMINE DISPLAY"
TELL DISPLAY "WORKSPACES 1 THROUGH 4, AND EDIT THE DATA IN"
TELL DISPLAY "EACH ONE."
TELL DISPLAY "WHEN YOU ARE FINISHED EDITING, HOME THE CURSOR"
TELL DISPLAY "IN EACH WORKSPACE."
TELL DISPLAY "TO RESUME THE EXECUTE FILE, PRESS 'COMMAND',"
TELL DISPLAY "'NEXT', 'RESUME', 'COMMANDS'."
TELL DISPLAY "YOUR EDITED DATA IN EACH WORKSPACE WILL BE STORED"
TELL DISPLAY "IN 4 SEPARATE FILES ON THE DISKETTE."
SUSPEND COMMAND FILE
PURGE FILE FILE1
PURGE FILE FILE2
PURGE FILE FILE3
PURGE FILE FILE4
DISPLAY WINDOW#1
COPY FILE FROM DISPLAY TO FILE1
DISPLAY WINDOW#2
COPY FILE FROM DISPLAY TO FILE2
DISPLAY WINDOW#3
COPY FILE FROM DISPLAY TO FILE3
DISPLAY WINDOW#4
COPY FILE FROM DISPLAY TO FILE4
DISPLAY WINDOW#1
TELL DISPLAY "I AM DONE; EXECUTE FILE COMPLETED."
EXIT COMMAND FILE
```

Execute File for Disc Drives

Step 5. Copy the command sequences from the display to disc filename "DOIT". Press [COMMAND], [COPY], ALL, [DISPLAY], [TO:file], DOIT, [RETURN].

Step 6. Start execution by pressing: [next], [EXECUTE], DOIT, [RETURN].

EXECUTING COMMANDS FROM A PROGRAM

The terminal I/O devices (alphanumeric and graphic displays, mini disc drives, printer, and devices connected to the HP-IB can be controlled from a computer through the use of two types of generalized escape sequences.

```
ESC,c <command sequence> ESC
ESC&p <parameter sequence>
```

Also, you can execute the ESC,c sequences from the mini disc drives.

The ESC,c method allows English-like commands to be used in the command sequence to perform a given function. It is recommended that you use ESC,c sequences, unless backward compatibility to HP 2645 or HP 2648 terminals is required.

TThe ESC&p method uses special codes of numerical and alphabetical characters to perform a given function. This method is compatible with HP 2645 and HP 2648 terminals.

The remainder of this section describes and gives examples of the commands using the ESC,c sequence, followed by descriptions and examples of the ESC&p sequences.

NOTE: In the remainder of this section, references to tapes should be interpreted as references to tape emulator files which are actually located on a mini-floppy disc. Refer to the paragraph "Cartridge Tape Emulation" in Section 6.

Using The ESC, c Generalized Escape Sequence

The command syntax is given in figure 5-3. An abbreviated form of the command syntax may be used by sending only the uppercase characters given in the command sequence. One of the words in braces "{ }" must be included in the command. Words in brackets "[ ]" may be omitted from the syntax.

NOTE: Many ESC,c commands will not be executed when incorporated as part of a file which is to be executed using the [READ] key. These commands are listed below:

Append	Execute	Rename	Tell
Compare	Exit	Report	Test
Copy	Find	Resume	Transfer
Create	Format	Rewind	Unpurge
Disable	Hello	Show	Verify
Edit	Mark	Skip	
Enable	Pack	Suspend	
	Purge		

In addition, the ESC&p sequence used with codes B, F, and M will not be executed under these circumstances.

Table 5-3 shows the command syntax abbreviations (and their meanings) in alphabetical order. (You may find this table helpful when debugging programs that contain the abbreviated syntax.)

APpend. The APpend command sequence allows you to add data at the end of an existing disc file. If the file is next to the free space (i.e., the last file on the volume), then it is extended until the APpend is completed. If the file is blocked by another file, the APpend command sequence copies as much data as possible, then stops with an "END OF DISC FILE" error message. The last line read will be lost.

Table 5-3. Command Syntax Abbreviations

All	Disable	Mark	SHow
AM	DISC#1{1 or 2}	Name	SKip
APpend	DISCS	Null	Source
Application	Edit [mode]	Numeric	Status [of]
Assign	Enable	PAck	SUspend
Assignments	End [of]	Page	Tapes
BYE	EXecute	PM	TELL
CALENDAR	EXIT	Printer	TErminal
CLOse	External printer	PRogram	Test
Command file	File	PURGE	TIme
COmpare	Find	Record [mode]	TRansfer
CONdition	FORmat	RENname	Window
Copy	Graphics	REPort	UNPurge
CReate	Header	RESume	VERify
CTU	HELLO	Rewind	Volumes
DIsplay	Hp-ib#<p>[#<s>	Right tape	
Data	[#<m>]]	SET	
DAtacomm	Left tape	SHared printer #<n>	
Date	Line		
Destination	LOG		

NOTE: The uppercase characters in each word are the abbreviation.

If you do not specify "from" or "to" devices in your command sequence, the currently assigned "Source" and "Destination" devices/files are used. (See the Assign command).

The defaults are: File, Source, Destination.

Multiple "to" devices/files may be specified; separate them with a comma (the data will be appended to each one in parallel).

If the display is specified as the "to" device, the data is "appended" at the current position of the cursor on the display (same as the copy command).

NOTE: If the "to" file is ASCII (not binary), any 8-bit codes are expanded to their representative display escape sequences prior to being sent to the disc file. Refer to "Data Operations" earlier in this section for further information on data operations using the APpend command.

Example: Append the data from the display (from the current cursor position to the end of display memory) to the disc file named "MEMO".

```
ESC,c APpend All [from] DIsplay [to] MEMO &
ESC,c AP A DI MEMO &
```

Example: Append the data from the disc file named "REPORT" to the display and Shared Printer #6.

```
ESC,c APpend File [from] REPORT [to] DIsplay,
     SHared printer#6 &
ESC,c AP F REPORT DI, SH#6 &
```

Assign. The Assign command sequence specifies which device(s) or disc files will be assigned to the logical filenames Source, Destination, LOG, Printer, CTU (cartridge tape unit), or your own logical filename. The default values are:

Source = Left tape
Destination = Right tape
LOG = Display
Printer = EXternal printer
CTUL = CTUL???
CTUR = CTUR???

Multiple devices may be assigned to Destination and LOG; commas are required to separate the device names. (Multiple devices may be assigned to Source, but when execution of a command containing or using the Assigned Source as a "from" device is attempted, thhe command will fail and error code 263, "TOO MANY DEVICES SPECIFIED" will be returned to the host if a REport Status command is sent from the host. Commands containing "Source" as a "to" device will execute, however.

Sometimes, it is very useful to use the Assign command to assign a disc file to a CTU Emulator file to permit execution of additional commands (such as, COmpare, SKip, etc.).

Example: Assign a disc file named "SALES" and the local printer as the Destination devices.

```
ESC,c Assign Destination [to] SALES,
     EXternal printer &
ESC,c A D SALES, EX &
```

Example: Assign the right tape and local printer as the Destination devices.

```
ESC,c Assign Destination [to] Right tape,
     EXternal printer &
ESC,c A D R, EX &
```

Example: Assign the display workspaces as the Source device, (The particular workspace (1, 2, 3, or 4) must be selected by the DIsplay command sequence.)

```
ESC,c Assign Source [to] DIsplay &
ESC,c A S DI &
```

Example: Assign the disc file "INVENTORY" on volume "QTR2" (which is on disc drive no. 2) as the source device.

```
ESC,c Assign Source [to] INVENTORY;QTR2 &
ESC,c A S INVENTORY;QTR2 &
```

Example: Assign the data communication line to the host computer as the destination for data.

```
ESC,c Assign Destination [to] DAtacomm &
ESC,c A D DA &
```

Example: Assign "PLOT" as a user-assigned name to the left tape.

```
ESC,c Assign Name PLOT [to] Left tape &
ESC,c A N PLOT L &
```

Example: Assign "PAPER" as a user-assigned name to the HP 7245 Plotter Printer connected to the HP-IB as device#4.

```
ESC,c Assign Name PAPER [to] Hp-ib#4 &
ESC,c A N PAPER H#4 &
```

BYE. The BYE command sequence terminates use of the current user-group assigned by the HELLO command sequence on the shared printer listings.

Example:

```
ESC,c BYE &
```

CALENDAR. The calendar command is used to display a one-month calendar for the specified date. The specified date is highlighted on the displayed calendar. If no date is specified, the current date is used (i.e. the date previously set with the SET Date command).

Example: Display a calendar with the current date highlighted.

```
ESC,c calendar ↵
```

Example: Display a calendar for June 17, 1948.

```
ESC,c calendar 6/17/1948 ↵
```

Example: Display a calendar for the 15th day of the current month.

```
ESC,c CALENDAR 15cr
```

CLOse. The CLOse command turns off any of the terminal display lines (message, command, or softkey labels) on the screen. (For turn-on, see DIsplay command.)

Example: Turn off the message line on the display.

```
ESC,c CLOse Window#5 ↵
ESC,c CLO W#5 ↵
```

Example: Turn off the command line on the display.

```
ESC,c CLOse Window#6 ↵
ESC,c CLO W#6 ↵
```

Example: Turn off the softkey label line on the display.

```
ESC,c CLOse Window#7 ↵
ESC,c CLO W#7 ↵
```

COmpare. The COmpare command sequence makes a byte-by-byte comparison of each record on the specified disc files and/or devices. If a compare is not made in any record, a message will appear on the message line. The types of messages are as follows:

```
"DIFFERENT LENGTH RECORDS, RECORD#<N>"
"DIFFERENCE IN BYTE#<n>, RECORD#<r>"
"DIFFERENCE IN RECORD TYPE, RECORD#<n>"
```

Nothing is sent to the computer system to indicate successful or unsuccessful comparison; however, the Report Status command can be used to determine the completion status of the command (Refer to "Command Status" in Section 10).

After a mismatch is found, the comparison may be continued for CTU emulator files on the disc by reissuing the command. (A disc file may be converted to a CTU emulator file by using the Assign command to assign the file to either CTUL or CTUR.)

The defaults are: File, Source, Destination.

Refer to "Cartridge Tape Emulation" in Section 6 for operation of this command on tape-emulated files.

Entire disc volumes may be compared by using the COmpare Volumes command sequence. (This requires two disc drives, one for each volume.)

NOTE: Unlike data types can't be compared. For example, graphics data can't be compared to alphanumeric data. Refer to "Data Operations" earlier in this section for further information on data operations using the COmpare command.

Example: Compare the disc volume "FEB80" with the disc volume "FEB80A" mounted in disc drive no. 2.

```
ESC,c COmpare Volume FEB80 to DISC#2 ↵
ESC,c CO V FEB80 DISC#2 ↵
```

Example: Compare disc file "MEMO" to disc file "MEMO2" on the same diskette which is mounted in disc drive no. 1.

```
ESC,c COmpare File MEMO [to] MEMO2 ↵
ESC,c CO F MEMO MEMO2 ↵
```

Example: Compare the data in the set of CTU emulator files assigned to CTUL to those CTU emulator files assigned to CTUR.

```
ESC,c COmpare All [of] Left tape [to] Right tape ↵
ESC,c CO A L R ↵
```

Example: Compare the file of the source device to the destination device. (The defaults are: File, Source, Destination.)

```
ESC,c COmpare ↵
ESC,c CO ↵
```

Example: Compare a line of the device with a user-assigned name of "INPUT" to the device with a user-assigned name of "OUTPUT".

```
ESC,c COmpare Line [of] INPUT [to] OUTPUT ↵
ESC,c CO L INPUT OUTPUT ↵
```

Copy Line, File, All

NOTE: Refer to Copy Volume command sequence for making duplicate copies of diskettes.

The Copy command sequence copies either 7-bit ASCII data or 8-bit binary data (a line, a file, or all) from the specified source to the specified destination. The defaults are Source, Destination, and File. If the computer system sent ↑,c Copy ↑, a file would be copied from Source to Destination. Multiple disc files or devices may be specified as destinations by separating the disc file or device names with a comma, or by equating the disc file or device names to Destination in the Assign command sequence.

The "from" file must exist on the selected volume. The volume name or device number (see "disc file name" in figure 5-3) must be specified as part of the file name if the volume is mounted on disc drive no. 2 in a two-drive system.

The "to" file must not exist on the selected volume. The device number (disc #1 or disc #2) must be specified as part of the file name if the volume is mounted on disc drive no. 2 in a two-drive system.

The file will be copied without need to specify the amount of disc space required for a new file. The new file will be added to the end of the currently used space of the disc volume.

The name of the new file will be appended to the directory so that a SHOW FILES command sequence will display the file name.

When copying a binary disc file to another disc file, the file type remains binary; it does not change to ASCII, and all 8-bit codes are copied.

COPY ALL and COPY FILE respond the same for a disc file. The operation is complete when the end-of-file is reached.

If an error occurs during the copy (such as "Read Fail"), the files will be left open while the error message is displayed. When a subsequent I/O command is issued, the files that were left open when the error occurred are closed; then the files required by the new I/O command are opened.

Refer to "Cartridge Tape Emulation" in Section 6 for operation of this command on tape-emulated files.

NOTE: Refer to "Data Operations earlier in this section for further information on data movement using the Copy command.

Example: Copy a file from the Source device to the Destination device.

```
↑,c Copy ↑
↑,c C ↑
```

Example: Copy all data on the Source device (assume the display to the local printer and a new disc file "EXPENSES".

```
↑,c Copy all [from] Source [to] External
     printer, EXPENSES ↑
↑,c C A S EX, EXPENSES ↑
```

Example: Copy all data on the Source device (assume the display to the local printer and the file currently "selected" on CTUR (the set of files selected for right tape CTU emulation).

```
↑,c Copy All [from] Source [to] External
     printer, Right tape ↑
↑,c CN A S EX, R ↑
```

Example: Copy the contents of the disc file "ORDERS" to the HP-IB device#4 (e.g., HP 7310A Printer). Data must be ASCII to copy to a non-raster dump type printer.

```
↑,c Copy File [from] ORDERS [to] Hp-ib#4 ↑
↑,c C F ORDERS H#4 ↑
```

Example: Copy the contents of the left tape (CTUL tape emulation file currently selected) to the HP-IB device #4 (e.g. HP 7310Aaprinter). Data must be ASCII.

```
↑,c Copy FIle [from] Left tape [to] HP-IB#4 ↑
↑,c C F L H#4 ↑
```

COPY VOLUME.

A sector for sector copy of the "from" volume is made on the "to" disc device. The "to" disc must not have a volume name, but it must be HP-formatted. (Refer to the FORmat Volume command). If any files are opened on the "from" volume, this command will not start. This command cannot be executed in a single flexible disc drive system.

Example: Copy the volume "SCHEDULES" to a new volume mounted on disc drive no. 2.

```
↑,c Copy Volume from SCHEDULES to DISC#2 ↑
↑,c C V SCHEDULES D#2 ↑
```

CReate.

The CReate command enables preparation of either a disc (volume) or file area on disc for data storage.

NOTE: Any files or volumes created before a date is set will have no date/time stamp. Since no default date is provided, the SET Date command must be used to set a date if a time/date stamp is desired on created files and volumes.

CReate File. This command allows you to reserve an area on the disc volume for future use. When the command is issued, an End-of-File mark is written at the beginning of the file. If the "BYTES" entry is not specified, 80 byte lines are assumed. If the "LINES" entry is not specified, one line (80 bytes long) is assumed. If "TYPE" is not specified, ASCII is assumed.

The number of sectors allocated to the file is calculated by the following formula:

```
(<n> lines x <m+2 bytes)
------------------------    ~ no. of sectors
          256
```

The overhead is comprised of two bytes per record.

For example: Create a file of 56 lines with 80 bytes per line.

```
<n> lines = 56
<m+2> bytes = 82 (2 additional bytes per line
                  for overhead)
(56x82)
-------   ~ 18 sectors
  256
```

The free space pointer for the volume is updated when execution of the command is completed. Errors are returned if the file name already exists or if the requested size exceeds the available disc space.

Example: Create an ASCII file with 56 lines of 80 bytes each, and name the file "SUE".

```
:c CReate File SUE [of] 56 lines [of]
     80 bytes [of] Type Ascii ↵
:c CR F SUE 56 L 80 B T A ↵
```

The SHow Files command lists the CReated file "SUE", 18 sectors long:

```
Volume = LAB

  File   File    Start     Size       Create          Create
  Name   Type    Address   (sectors)  Date (M/D/Y)    Time
  ----   -----   -------   ---------  ------------    -------
  SUE    Ascii    32         18        5/15/80        2:48 PM
```

CReate Volume. If an unformatted disc is in the drive when the CReate command is issued, a "FORMAT" occurs first with the default stagger spacing (9). (A different stagger can be selected using the FORmat command.) Then the CReate Volume command writes a volume label on the disc, and allocates space for the directory. No volume label can exist on the disc when the CReate Volume command is issued; existence of such a label causes the command to fail. (Error code 462, "CREATE VOLUME not allowed", will be sent to the host if a REPort Status [of] Command is issued.) To change the name of a labelled volume, refer to the REName Volume command.

Example: CReate a volume named "WARREN" on disc drive no. 2.

```
:c CReate Volume WARREN [on] DISC#2 ↵
:c CR V WARREN DISC#2 ↵
```

DIAL. The DIAL command uses the 13265A Modem Pod to dial the requested telephone number. The command looks in the Assign table to find the specified name. If the name matches the one in the Assign table, then the string value associated with the name is used for dialing. If the name is not found, then it is dialed according to the conventions established for converting letters to their equivalent numbers as shown on a telephone dial.

The string may contain anything but only certain characters are acted upon:

Numeric (0-9)	These are acted upon immediately (0 equals 10 pulses).
Alpha (A-Y, a-y)	These are dialed immediately (accordingly to the telephone dial convention).
@	One-second delay. This is used as a wait mechanism for slow-connecting telephone equipment (10 pulses per second). You may concatenate @s together to increase the wait time. They may be placed anywhere within the string.
>	Use fast dialing rate (20 pulses per second) instead of slow dialing rate. This character should appear as the first character in the string. The default dialing rate is slow; it is set too fast when the ">" character is encountered in the string.
~	Inform user that dialing is completed. Wait for RETURN key to disconnect modem from the telephone line. This character should be last in the string.

The DIAL command allows you to use parentheses in area codes and a hyphen in the seven-digit telephone number. However, these characters are not required; you may enter a number "4087358200" instead of "(408) 735-8200". (Blanks may be used, also.)

Example: Dial the telephone number 735-8200.

```
:c DIAL 735-8200 ↵
```

Example: Dial "POPCORN". (The numeric equivalents of the letters are dialed, according to the phone dial convention - 767-2676.)

```
:c DIAL POPCORN ↵
```

Example: Assign the name "COMPUTER" to the telephone number "735-9865", then dial the number.

```
:c Assign Name COMPUTER [to] 735-9865 ↵
:c DIAL COMPUTER ↵
```

Example: Dial a long distance telephone number requiring an area code and delays for slow-connecting telephone equipment. The number is "(408) 735-7535".

```
ESC,c DIAL (4@0@8)@@ 7@3@5@-@7@5@3@5 CR
```

ENABLE/DISABLE. (EDIT MODE, RECORD MODE, VERIFY MODE AND THE GRAPHICS/ NUMERIC KEYPAD). The Enable and Disable command sequences turn on and off Edit Mode, Record Mode, and Verify Mode and convert the graphics/numeric keypad to numeric operation from graphics operation and to graphics operation from numeric operation.

Edit Mode. In local mode (REMOTE key up), the assigned source and Destination devices are the source and destination for data respectively. The display workspace presently on the screen displays the data read from Source. In Edit mode, the [READ] key causes data to be copied from the source to the display (press the [RETURN] key to stop the operation at any point). The [RECORD] key copies remaining data on the display and source to the destination and terminates Edit mode.

Also see "EDIT FILE" command sequence.

In Remote mode (REMOTE key down), Data Logging Mode is enabled.

Example: Turn on Edit Mode

```
ESC,c Enable Edit [mode] CR
ESC,c E E CR
```

Record Mode. Enabling Record Mode has the same effect as pressing the [RECORD] key. If the terminal is in Remote mode (REMOTE] key down), data that is present on the datacomm line is copied to the assigned Destination device(s). The default destination is the right tape. If the terminal is in Local Mode ([REMOTE] key up), data on the display is copied to the assigned Destination device(s). The default destination is the right tape (CTURxxx for CTU emulated files).

Example: Turn off Record Mode.

```
ESC,c Disable Record [mode] CR
ESC,c D R CR
```

Verify Mode. The Enable/Disable Verify command sequence turns the write-backspace-read function on and off. This function assures the integrity of data sent to a disc file by writing the physical data record, backspacing to the beginning of the record, then reading the record and comparing it to the original record.

The Verify Mode is in effect until turned off.

Example: Turn on Verify Mode.

```
ESC,c Enable Verify [mode] CR
ESC,c E V CR
```

Graphics/Numeric Keypad. The graphics/numeric keypad can be converted to numeric operation from graphics operation or vice-versa.

Example: Convert the keypad from graphics operation to numeric operation.

```
ESC,c Enable Numeric [mode] CR
ESC,c E N CR
```

or

```
ESC,c Enable Numeric [keypad] CR
ESC,c E N CR
```

Example: Convert the keypad from numeric operation to graphics operation.

```
ESC,c Enable Graphics [mode] CR
ESC,c E G CR
```

or

```
ESC,c Enable Graphics [keypad] CR
ESC,c E G CR
```

DIsplay. The DIsplay command sequence controls the four display workspaces and the three terminal control lines (message, command, and softkeys). Also, the DIsplay command sequence may be used to specify the display workspace (1, 2, 3, or 4), message line (5), or command line (6) as the destination for data that is to follow. (More information on the display workspaces and control lines is given in Section 2.)

Example: Send "Insert 'MAILING LIST' diskette into disc drive.", in inverse video, to the message line. (To turn off the message line, use the CLOse command described previously.)

```
ESC,c DISplay Window#5 CR ec&dBInsert
    'MAILING LIST' diskette into disc drive CR

ESC,c DI W#5 CRec&dBInsert
    'MAILING LIST' diskette into disc drive CR
```

When the message line is opened, the next 80 ASCII characters are sent to it for display. Less than 80 characters may be specified by inserting a CR at the end of the string.

Example: Turn on Memory Lock in workspace 3, clear workspace 1, and display workspace 2 on the screen.

```
ESC,c DIsplay Window#3 ↵ EcI
ESC,c DIsplay Window#1 ↵ EcH EcJ
ESC,c DIsplay Window#2 ↵

ESC,c DI W#3 ↵ EcI
ESC,c DI W#1 ↵ EcH EcJ
ESC,c DI W#2 ↵
```

Example: Display the command line. (This command does not execute when issued from a computer system.)

```
ESC,c DIsplay Window#6 ↵
ESC,c DI W#6 ↵
```

EDIT File. The EDIT File command sequence simplifies entering Edit Mode. The Source and Destination files/devices may be specified in the command syntax rather than separately as in Enable Edit and Assign command sequences.

This command does not alter the "Source" and "Destination" assignments in the Assign Table.

The defaults are Source and Destination files/devices as specified by the Assign command.

NOTE: It is not possible to edit two disc files at the same time, but it is possible to edit to both a disc file and another "source/destination" at the same time.

Example: Turn on Edit Mode using file name "CHAPTER4" as the source and file name "Chapter4A" as the destination.

```
ESC,c EDIT File [from] CHAPTER4 [to] CHAPTER4A ↵

ESC,c EDIT F CHAPTER4 CHAPTER4A ↵
```

Example: Turn on Edit Mode using file name "JUL_STOCK" as the source and default destination file/device.

```
ESC,c EDIT File [from] JUL_STOCK ↵
ESC,c EDIT F JUL_STOCK ↵
```

Example: Turn on Edit Mode using default source and destination files/devices.

```
ESC,c Edit File ↵
ESC,c EDIT F ↵
```

or

```
ESC,c Enable Edit ↵
ESC,c E E ↵
```

Example: Turn off Edit Mode.

```
ESC,c Disable Edit [Mode] ↵
ESC,c D E ↵
```

EXecute. The EXecute command sequence allows command sequences to be executed from a disc file or device rather than from the computer system. Each command sequence must be on a separate line or record. If user intervention is required at any point in the execution (such as inserting a disc), the SUspend command sequence may be used. (Also see "EXIT" and "RESume".)

Example: Execute a list of commands stored on disc and assigned to CTUL.

```
EXecute Left tape ↵
ESC,cEX L ↵
```

The commands to be executed do the following:

- Read file 2 (assigned to CTUR) to display workspace 1.
- Suspend execution for user editing of the displayed data.
- Copy the edited data on the display to the printer.

The list of commands assigned to CTUL are:

```
Find File 2 [on] Right tape ↵
DIsplay Window#1 ↵
Copy FIle [from] Right tape [to] DIsplay ↵
SUspend Command file ↵
```

(user must use RESUME command here.)

```
Copy File [from] DIsplay [to] EXternal printer ↵

EXIT Command file ↵
```

or in abbreviated form:

```
F F 2 R ↵
DI W#1 ↵
C F R DI ↵
SU C ↵
```

(user must use RESUME command here.)

```
C F DI EX ↵
EXIT ↵
```

EXIT. The EXIT command sequence terminates execution from a file. (See "EXecute" above.)

Find. The Find command sequence is used with CTU emulation files. It positions the device (CTU Emulator files on the disc) to an absolute or relative (+, -) file number or to the end-of-data mark. Some printers position to the top-of-form when a Find file command is sent. Disc files are located by their file name, not a number.

Refer to "Cartridge Tape Emulation" in Section 6 for operation of this command on tape-emulated files.

Example: Find file 4 on the left tape (CTUL), then position the left tape forward 6 lines.

```
ESC,c Find File 4 [on] Left tape ESC
ESC,c Skip +6 [lines]  [on] Left tape ESC
or
ESC,c F F 4 ESC
ESC,c SK +6 ESC
```

Example: Find the end-of-data mark on the right tape.

```
ESC,c Find End [of] Data [on] Right tape ESC
or
ESC,c F E D R ESC
```

FORmat Volume. The FORmat Volume command sequence records the soft sector information that is required to make a diskette compatible with the terminal file system. A new diskette must be formatted prior to use. If the command sequence is issued to a floppy disc that is already formatted, any data on the disc will be destroyed. The message "Disc FORMAT in progress" is displayed in the message window while the command is executing.

If the stagger is not specified, then the default stagger (9) is used. The maximum stagger is 15.

During the FORmatting operation, any defective sector causes the entire track containing the sector to be marked defective. If there are a sufficient number of cylinders available for use (33 in the discs), the disc is pronounced fit for use. If an insufficient number of cylinders are found after four attempts to FORmat an acceptable disc, the disc is rejected.

Upon successful completion, a disc VERify Volume command is executed automatically. The message "Disc VERIFY in progress" is displayed in the message window while the command is executing.

Example: Format a floppy disc on Disc Unit#1.

```
ESC,c FORmat Volume [on] DISC#1 ESC
ESC,c FOR V DISC#1 ESC
```

Hello. The HELLO command sequence allows the user to assign a "usergroup" to the listings printed by the shared printer. Up to eight characters may be assigned to each field.

Example: Assign "PROJECT3.CHEMLAB" to be printed on all shared printer listings.

```
ESC,c HELLO PROJECT3.CHEMLAB ESC
```

Mark File. The Mark File command sequence closes the specified file on the diskette or writes a file mark on the specified CTU-Emulator set of files (cartridge tape). Some printers position to top-of-form when a Mark File command is sent.

It is not necessary to issue a Mark File command to close standard disc files (i.e., non-CTU Emulator files). They are closed automatically after the data transfer command has written data to the file. However, CTU Emulator files must be closed after data is written to them. Either a Mark File command or any tape movement command that "positions the tape" backward (REwind, Find File, or SKip Lines) will close a CTU Emulator file. If the terminal is in Forms Mode, the CTU Emulator file is closed automatically after a file is recorded.

Use of the Mark File command when the CTU Emulator is positioned at the end of a file will create an empty file. Also, when the HP 3000 computer proggram FCOPY is used to write a CTU Emulator file on one of the terminal diskettes, it automatically creates a new empty file by sending two Mark File commands; the first closes the current file and the second creates the new empty file.

Example: Write a file mark on the right tape.

```
ESC,c Mark File Header [on] Right tape ESC
ESC,c M F H R ESC
```

PAck. The PAck Volume command sequence deletes the files that have been purged (see Purge command sequence). First, the directory is compacted, and all the purged dirrectory entries are removed. Next, the data areas associated with unpurged files are moved into those areas vacated by the purged files. Lastly, the free space pointer for the volume is updated.

The PACK operation may take several minutes. If there are open files on the volume, the PAck command will not start.

The message "Disc PACK in progress" is displayed in the message window to remind the user that the operation is in progress. This message also contains information about the current sector being copied from the current sector being copied to so that recovery is possible if a disc error occurs. When the message is followed by a single number, it indicates that the directory compaction is in progress. When two numbers appear, the actual movement of data files is occurring.

Example: PAck volume named "ACCNTS"

```
ESC,c PAck Volume ACCNTS ESC
ESC,c PA V ACCNTS ESC
```

PURGE File. The PURGE File command effectively eliminates the specified file (renders it inaccessible) except that the file remains on disc and can be restored to normal status with the UNpurge command.

If the file is currently open for access, the Purge command fails. If the file does not exist, or has been purged previously, an error code is returned.

If a disc containing purged files is used on any HP system other than an HP 2647F the new system may not recognize the purged condition of the purged files; it may consider them valid files, possibly resulting in an error condition. (If a volume is packed, this problem will not occur.)

Example: Purge the file named "JEFF".

```
:c PURGE File JEFF
:c PURGE F JEFF
```

PURGE Volume. The PURGE Volume command sequence deletes the volume label making all data on the disc inaccessible.

The destination volume must be purged before a FORmat or Copy Volume command will execute.

If the volume was purged accidentally, it may be restored by using the UNPurge Volume command. However, if the volume has been FORmatted after being purged, the volume cannot be restored; the data is lost.

Example: PURGE the volume named "DEBITS".

```
:c PURGE Volume "DEBITS"
:c PURGE V DEBITS
```

REName File. The REName File command sequence allows you to rename an existing file. The old file name is deleted from the directory and is replaced by the new file name. The file still occupies the same space on the disc (the new file is not copied to another point on the disc). The entry in the directory retains its original order in the list of files.

The command will not execute if you specify a volume name as part of the new file name (i.e., <new file name>;< volume name>); the error message "`Volume name not allowed`" will be displayed in the message window to indicate the improper syntax.

The command will not execute if the new file name already exits on the disc volume or if the file is open.

Example: Rename the file "UNITSOUT" to "SHIPMENTS".

```
:c REName File [from] UNITSOUT [to] SHIPMENTS
:c REN F UNITSOUT SHIPMENTS
```

Example: Rename the file "WORK" on the volume "OCT80", which is mounted disc drive no. 2, to "TASKS".

```
:c  REName File [from] WORK;OCT80 [to] TASKS
:c REN f WORK; OCT80 TASKS
```

Example: Rename the ACCOUNTS file to CTU Emulator file CTUR005.

```
:c REName File [from] ACCOUNTS [to] CTUR005
:c REN F ACCOUNTS CTUR005
```

REName Volume. The REName Volume command sequence allows you to rename an existing volume. The disc volume retains all the files, unpurged and purged, that are in the directory. The command will not execute if there are open files on the volume.

Example: Rename the volume "TOM" to "JERRY".

```
:c REName Volume [from] "TOM" [to] "JERRY"
:c REN V TOM JERRY
```

REPort. The REPort command sequence sends the successful/unsuccessful completion status of the previous command execution to the datacomm line. The status consists of a 5-digit number; the meaning of this number is given under the heading "Command Status" in Section 10 (Status).

If the last command was EXecute, the status of the successful/unsuccessful completion of the EXecute command is delayed until the EXecute file is terminated.

Example:

```
:c REPort Status [of] Command
```

or

```
:cREP S C
```

RESume. The RESume command sequence returns control to the command file, or the application, if it had been suspended. (Also see SUspend and EXecute.)

Example:

```
ESC,c RESume Command file CR
```
or
```
ESC,c RES C CR
```

REwind. Refer to "Cartridge Tape Emulation" in Section 6 for operation of this command on tape emulated files.

Example: Rewind the tape unit with the user-assigned device name "BILLING".

```
ESC,c REwind BILLING CR
```
or
```
ESC,cRE BILLING CR
```

SET (Time,Date). The SET Time command sequence allows the user to set the current time in hours, minutes, and seconds (optional). Once the time is set, it will not require resetting until the terminal is turned off. The ALARM command sequence uses this time as a reference.

Example: Set the terminal time to 10 hours, 15 minutes, 23 seconds AM.

```
ESC,c SET TIme 10:15:23 AM CR
ESC,c SET TI 10:15:23 AMCR
```

NOTE: Once set, the date stored cannot be eliminated except by a hard reset or turning the terminal off. Also, it cannot be changed except by using the SET DATE command with the format:

mm/dd/yyyy

An attempt to change the date by the string method has no effect.

The SET Date command sequence allows two methods of setting the date:

Method 1. You can specify the date by entering the appropriate digits for month, date, and year (for example, May 23, 1980 would be 5/23/80). One, two, or three numbers can be entered, separated by slashes (/), dashes (-), or commas (,). If only one is entered, it is taken to be the date, and the previously-set month and year are used. If two numbers are entered, they are taken to be the month and date and the previously-set year is used. If an invalid date is entered, an error message is generated and the previously-set date is used. The first Set Date command after a power on must have all three numbers provided since there are no previously-set numbers to use for the default values. Once set, the date is not changed by either a soft or hard reset.

The date set by this method is updated automatically each day by the timer that is set by the SET TIme command (i.e., at midnight). However, the date and time will require resetting if the terminal is turned off.

This form of the SET Date command is the only form recognized for changing the creation date of disc files and volumes. There is no default date set for power-on time and any files and volumes created before the date is set will not have a date/time stamp.

Method 2. Using the second method, you set the date by specifying any data string, up to 30 characters, enclosed in quotes. (For example, "Friday, May 23, 1980".) This method requires that the date be set each day (i.e., it is not updated automatically by the timer). Also, it will require resetting if the terminal is turned off or hard reset. The data string appears on the header and trailerpages of shared printer listings but is not used for the date/time stamp on files or volumes.

Example: Set the terminal time to 10 hours, 15 minutes, 23 seconds AM.

```
ESC,c SET TIme 10:15:23 AM CR
```
or
```
ESC,cSET TI 10:15:23 AMCR
```

Example: Set the terminal date to "FRIDAY, JULY 24, 1978".

```
ESC,c SET Date "FRIDAY, JULY 14, 1978" CR
ESC,cSET D "FRIDAY, JULY 14, 1978CR
```
or
```
ESC,c SET Date 7/14/78 CR
ESC,cSET D 7/14/78CR
```

SHow (Assignments, Tapes, Time, Date). The SHOW command sequence lists the device assignments, current terminal time, or current terminaldate string on the LOG device.

Example: Show the current device assignments for Source, Destination, LOG, and user-defined assignments.

```
ESC,c SHow Assignments CR
```
or
```
ESC,cSH ACR
```

The following listing is an example of the SHow Assignments listing.

```
USER NAME     CURRENT ASSIGNMENTS
-----------   -------------------
SOURCE        LEFT TAPE
DESTINATION   RIGHT TAPE
LOG           DISPLAY
PRINTER       EXTERNAL PRINTER
CTUL          CTUL???
CTUR          CTUR???
```

Example: Show the current terminal time.

```
↯,c SHow TIme ↯
```
or
```
↯,cSH TI↯
```

The following listing is an example of the SHow TIme command output.

```
11:09:19 AM
```

Example: Show current terminal date string.

```
↯,c SHow Date ↯
```
or
```
↯,cSH D↯
```

The following is a SHow Date command listing; the date was set by the SET Date <month>/<date>/<year> method.

```
WED, MAY 7, 1980
```

The following is a SHow Date command listing; the date was set by the SET Date "<string>" method.

```
Wednesday, May 7, 1980
```

SHow Files. The SHow Files command sequence generates a listing (on the LOG device) of the file names on the on-line volumes. The Show Files command lists the files on either the disc (disc 1), the optional external disc (disc 2), or both. If the volume or device is specified in the command, only the files on the specified volume or disc are listed. The simple Show Files command, not followed by any specification, lists the files on both discs. If there is no on-line volume whose files can be listed, an error message is displayed. Possible error messages are "`Bad disc format`" (code 500), "`No disc in drive`" (code 497), or "`Volume name not found`" (code 453).

The listing contains the file name, file type (binary or ASCII), starting address on the disc volume, the length of the file in sectors, and the create date, as shown below.

Listing of a diskette mounted in disc drive no. 1:

```
Volume - Murray

File      File    Start     Size      Create       Create
Name      Type    Address   (sectors) Date(M/D/Y)  Time
--------  ------  --------  --------- -----------  --------
EXAMPLE   Ascii      32         8      11/ 1/79     11:32 AM
FORM      Binary     40         4      11/ 1/79      1:12 PM
CTUROC1   Ascii      44         1      11/ 1/79      6:02 AM
LOG       Ascii      45         3      12/12/79     10:00 AM
COMPARE   Ascii      48         4      12/ 3/79      4:06 PM
HARRY     Ascii      52         4      12/ 3/79      9:51 AM
SPECS     Ascii      56        13      12/14/79     10:10 AM
CTUL001   Ascii      69         1       1/22/80     12:21 PM
CTUL002   Ascii      70         1       1/22/80      7:05 AM
CTUL003   Ascii      71         1       1/22/80     11:53 AM
CTUL004   Ascii      72         1       1/22/80      8:31 AM
CTUL005   Ascii      73         1       1/22/80      3:45 PM
CTUR001   Ascii      74         2       1/23/80      2:16 PM
CTUR002   Ascii      76         2       1/23/80 w   11:23 AM
CTUR003   Ascii      78         2       1/23/80      3:19 PM
CTUR004   Ascii      80         2       1/23/80      1:49 PM
CTUR005   Ascii      82         2       1/23/80      1:13 PM
```

Listing of a diskette mounted in disc drive no. 2:

```
Volume - Cox

File      File    Start     Size      Create       Create
Name      Type    Address   (sectors) Date(M/D/Y)  Time
--------  ------  --------  --------- -----------  --------
DATASHT   Ascii      32        236      8/15/80      2:21 PM
FIELDTM   Ascii     268        584      8/14/80     11:02 AM
FORECAST  Ascii     852         47      9/10/80      2:31 PM
```

Pressing the RETURN key aborts the SHow Files command for the current volume. In order to terminate the SHow Files command for subsequent volumes, the RETURN key must be pressed for each subsequent volume.

The file name in the command sequence may include an "*" or a "?". An "*" substitutes for any number of characters, including "0"; a "?" indicates one character exists at the spot in the file name occupied by the "?".

For example, all files containing the letters "`it`" in their file name can be displayed by specifying "`*it*`" as the file name in the command sequence. All files ending in "`it`" will be displayed if "`*it`" is specified as the file name. All files beginning with "`it`" will be displayed if "`it*`" is specified as the file name.

If "`???it*`" is specified, all files with a name in which "`it`" appears preceded by three characters will be displayed. If "`it???`" is specified, all files with names starting with the characters "it" and consisting of five characters will be displayed.

Example: List all files beginning with "CTU".

```
↯,c SHow Files CTU* ↯
↯,c SH F CTU*↯
```

```
Volume - Murray

File      File    Start     Size      Create       Create
Name      Type    Address   (sectors) Date(M/D/Y)  Time
--------  ------  --------  --------- -----------  --------
CTUL001   Ascii      69         1       1/22/80      7:20 AM
CTUL002   Ascii      70         1       1/22/80      7:43 AM
CTUL003   Ascii      71         1       1/22/80      9:12 AM
CTUL004   Ascii      72         1       1/22/80     11:35 AM
CTUL005   Ascii      73         1       1/22/80      3:02 PM
CTUR001   Ascii      74         2       1/23/80      8:13 AM
CTUR002   Ascii      76         2       1/23/80     10:41 AM
CTUR003   Ascii      78         2       1/23/80     11:45 AM
CTUR004   Ascii      80         2       1/23/80      2:09 PM
CTUR005   Ascii      82         2       1/23/80      4:43 PM
```

Example: List only the third CTU emulator files (file 3 for the left and right tape).

```
↯,c SHow Files CTU?003 ↯
↯,cSH F CTU?003↯
```

```
Volume - Murray

File      File    Start     Size      Create       Create
Name      Type    Address   (sectors) Date(M/D/Y)  Time
--------  ------  --------  --------- -----------  --------
CTUL003   Ascii      71         1       1/22/80      9:34 AM
CTUR003   Ascii      78         2       1/23/80     11:08 AM
```

Example: List all files that have an "M" in their name.

```
ℓc SHowwFiles *M* ↵
ℓcSH F *M*↵

Volume - Murray

File      File     Start    Size       Create       Create
Name      Type     Address  (sectors)  Date(M/D/Y)  Time
--------  -------  -------  ---------  -----------  --------
EXAMPLE   Ascii       32          8    11/ 1/80     10:38 AM
FORM      Ascii       40          4    11/ 1/79      8:01 AM
COMPARE   Ascii       48          4    12/ 3/79      2:21 PM
```

If no file name is specified, then all files are listed; this is equivalent to:

```
ℓc SHow Files ↵or ℓcSH F↵
```

SHow Purged. The SHow Purged command sequence generates a listing of the purged file names (on the LOG device) on each of the on-line volumes. The listing has the same format as the SHow Files command sequence. This command allows you to locate a file that you may want to UNPurge. The "*" and "?" may be used in the <file name> to locate certain file names as described in SHow Files.

Example: List the purged files which are on the current on-line volume.

```
ℓc SHow PUrged ↵
ℓcSH P↵
```

Example: List the purged file "LAB STAFF", if present on the on-line volume.

```
ℓc SHow PUrged LABSTAFF ↵
ℓcSH P LABSTAFF↵
```

SHow Volumes. The SHow Volumes command sequence generates a listing (on the LOG device) of the shared peripherals on the HP-IB and the names of the local disc volumes.

The shared printer devices on the HP-IB are listed only if they can "self-identify", and the terminal's firmware recognizes the identity code, such as:

- HP 2631B option 046 Printer
- HP 2631A Printer
- HP 9871A Printer.

Other devices on the HP-IB are listed by "Hp-ib device <HP-IB address>" if they are currently being controlled by a terminal on the network. If any of these devices is not being controlled by a terminal it will not be listed in the SHow Volumes table. Some of these devices are listed below.

- HP 7245A Plotter Printer
- HP 9872A Plotter
- HP 7310A Printer
- HP 9874A Digitizer
- HP 9876A Printer

NOTE: The plotters are listed here because one or more HP 2647F Intelligent Graphics Terminals may be connected to the network and accessing one of the plotting devices. (If any device is being controlled by any terminal on the network, it will be shown in your terminal's volume table.)

The local disc volumes are listed with their names, type of disc, unit number, their current values for start of free space, and the amount of free space left. If any volume is identified as `nolabel`, `nogood`, or `nonfmt`, the start of free space is specified as zero (0) because the volume is not accessible and any free space specification may have no meaning.

If the listing reveals "`nonfmt`" for a volume, then the volume must be formatted and given a name before it can be used. (See CReate Volume and FORmat Volume commands.)

Example: List the devices on the HP-IB and the currently on-line volume (diskette).

```
ℓc SHow Volumes ↵
ℓcSH V↵

VOLUME NAME      HP-IB ADDRESS
---------------  -------------
HP-IB device     6
Shared Printer   7

Local   Disc       Start of    Space  Sector   Create          Create    Write
Volume  Type#Unit  Free Space  Left   Stagger  Date(M/D/Y)     Time      Protect
------  ---------  ----------  -----  -------  -----------     --------  -------
MURRAY  mfD#1      40          1016   9        9/23/81         8:33 AM   NO
```

SKip. The SKip command sequennce positions a device to a relative line, top-of-form, or beyond an end-of-data mark.

See "CTU Emulation" in Section 6 for operation of this command sequence.

Example: Position the CTU emulator file MEMO (assigned to CTUL as MEMO???) backward six lines. (This assumes the file is advanced six or more lines past the beginning of the file.)

```
ℓc SKip -6 <lines> <on> MEMO ↵
```

or

```
ℓcSK -6 MEMO↵
```

Example: Position the shared printer to next top-of-form.

```
ESC,c SKip Page <on> SHared printer#5 ENTER
```
or
```
ESC,cSK P SH#5ENTER
```

SUspend. The SUspend command sequence allows user intervention when executing command sequences from a file or application. A RESume command sequence returns control to the file containing the command sequences. An EXIT command sequence terminates execution. (See EXecution command.)

TEll. The TEll command sequence sends up to 80 characters to the specified device or a terminal in a network. The character string must be enclosed in quotes. Consecutive blanks in the string are reduced to a single blank.

Example: Send "This is Display Workspace 3" to workspace 3.

```
ESC,c DIsplay Window#3 ENTER
ESC,c TEll DIsplay "This is Display
     Workspace 3" ENTER
```
or
```
ESC,cDI W#3ENTER
ESC,cTE DI "This is Display Workspace 3" ENTER
```

Test. The Test command sequence tests the following:

- terminal (excluding disc drives and data paths)
- Disc drives (requires a diskette on which to write).
- datacomm line (the test connector must be installed — see "Data Communications Self Test" in Section 13).
- Data path to a specified HP-IB device.
- Data path to a specified terminal in a network.

Further explanation of self tests, interpretation of results, and appropriate action is given in the "Self Test" section.

Terminal. The terminal test verifies the operation of the terminal (such as the display, ROM, character sets, etc.)

Example: Test the terminal.

```
ESC,c Test ENTER
```
or
```
ESC,cTENTER
```

Disc Drives. The disc drive test performs a non-destructive test on the controller PCA, writes and reads two 256-byte records of AA/HEX (one on sector 1024 and one on sector 1040); then, writes and reads two 256-byte records of 55/HEX (one on sector 1024 and one on sector 1040). Following execution of the self-test, the terminal displays a message on the screen to indicate whether or not the disc has successfully completed the test or an error occurred. The possible messages are:

```
Disc self test ok
No disc controller attached
Disc self test fail, unit is <u>,
  Test is <t>, Subtest is <s>, Head is <h>
```

(See the Self Test Section for further information on the disc drive test.)

Tape Units. Refer to "Cartridge Tape Emulation" in Section 6 for operation of this command on tape-emulated files.

Example: Test the CTU emulated files.

```
ESC,c Test Tapes ENTER
```
or
```
ESC,cT TENTER
```

Datacomm. The datacomm test verifies the data communications PCA in terminal. The test connector, part no. 02645-60002, must be innstalled on the PCA to run the test. (See the Self Test section for further information.)

Example: Test the data communcations PCA.

```
ESC,c Test DAtacomm ENTER
```
or
```
ESC,cT DAENTER
```

Terminal#<r>. The data path between terminals in a network may be tested by specifying the assigned terminal number. The terminal number is its address set by switches B0 through B4 on the HP-IB/ROM PCA.

Example: Test terminal#5

```
ESC,c Test TErminal#5 ENTER
```
or
```
ESC,cT TE#5ENTER
```

HP-IB. The HP-IB/ROM PCA in the terminal may be tested. (See "Self Test", Section 13, for further information.)

Example: Test the HP-IB circuits in the terminal.

```
ESC,c Test Hp-ib RET
```
or
```
ESC,cT H RET
```

TRansfer. The TRansfer command sequence copies 8-bit binary data (a line, a file, or all) from the specified source to the specified destination. The defaults are Source, Destination, and File. If the computer system sent `ESC,c TRansfer <RET>`, a file would be copied from Source to Destination. Multiple devices may be specified as destinations by separating the device names with a comma and a space, or by equating the devices to Destination in the Assign command sequence.

NOTE: Only printers which accept raster dump data can be used as destinations for graphics data. Refer to "Data Operations" earlier in this section for further information on transfer of data using the TRansfer command.

Refer to "Cartridge Tape Emulation" in Section 6 for operation of this command on tape-emulated files.

Example: Copy a file from the Source device to the Destination device.

```
ESC,c TRansfer RET
```
or
```
ESC,cTR RET
```

Example: Copy all data on the Source device (assume the display to the local printer and the right tape).

```
ESC,c TRansfer All [from] source [to] EXternal
      printer, Right tape RET
```
or
```
ESC,cTR A S EX,R RET
```

Example: Copy the contents of disc file "DIRK" to the HP-IB device#7 (e.g. HP 2631G Printer).

```
ESC,cTRansfer file [from] DIRK [to] Hp-ib#7 RET
```
or
```
ESC,cTR F DIRK H#7 RET
```

UNPurge FILE. The UNPurge File command sequence restores a previously Purged disc file, providing that a "PACK" has not occurred since the file was purged. PUrged disc files may be listed by issuing a SHow Purged command sequence to the mounted disc volume.

If a purged file name is the same as one of the unpurged disc files on a volume, the UNPurge command will not execute because this would create two files with the same name. You may RENname the unpurged file, then issue an UNPurge command.

Example: List the purged disc files of a volume, and unpurge one of the files.

```
ESC,c SHow Purged RET
```
or
```
ESC,cSH P RET
```
```
Volume - Murray

File     File    Start    Size       Create       Create
Name     Type    Address  (sectors)  Date(M/D/Y)  Time
-------  ------  -------  ---------  -----------  --------
BILLING  Purged     32        52     11/ 1/79      9:09 AM
FORM2    Purged     84         4     11/ 1/79      1:47 PM
```

Now let's unpurge "BILLING".

```
ESC,c UNPurge File BILLING RET
```
or
```
ESC,cUNP F BILLING RET
```

Example: List the unpurged and purged files of a volume. Rename the unpurged file that has the same name as the purged file. Unpurge the purged file.

(1)
```
ESC,c SHow Files RET
```
or
```
ESC,cSH F RET
```
```
Volume - Murray

File     File    Start    Size       Create       Create
Name     Type    Address  (sectors)  Date(M/D/Y)  Time
-------  -----   -------  ---------  -----------  --------
EXAMPLE  Ascii      40         8     11/ 1/79     11:21 AM
FORM2    Binary     52         4      1/ 4/80      2:55 PM
CTUR001  Ascii      56         2      1/ 7/80      9:31 AM
LOG      Ascii      58         4     1/12/80      7:59 AM
COMPARE  Ascii      62         4      2/ 3/80      3:25 PM
```

(2)
```
ESC,c SHow Purged RET
```
or
```
ESC,cSH P RET
```
```
Volume - Murray

File     File    Start    Size       Create       Create
Name     Type    Address  (sectors)  Date(M/D/Y)  Time
-------  -----   -------  ---------  -----------  --------
EXAMPLE  Ascii      32         8     11/ 1/79     10:24 AM
FORM     Binary     48         4     11/ 1/79      8:33 AM
```

(3)
```
ESC,c REName File [from] EXAMPLE [to] EXAMPLE1 RET
```
or
```
ESC,cREN F EXAMPLE EXAMPLE1 RET
```

(4)
```
ESC,c UNPurge File EXAMPLE RET
```
or
```
ESC,cUNP F EXAMPLE RET
```

UNPurge VOLUME. The UNPurge Volume command restores a previously PURGEd volume provided neither the FORmat command nor the PAck command has been used on the volume after the PURGE command was performed.

Example: List a PURGEd volume and UNPurge it.

```
ESC,c SHow Volumes ESC%
or
ESC,cSH V;
```

```
Local   Disc       Start of    Space  Sector   Create        Create    Write
Volume  Type#Unit  Free Space  Left   Stagger  Date (M/D/Y)  Time      Protect
------  ---------  ----------  -----  -------  ------------  --------  -------
nolabl  mfD#1             0   1056       9                            NO
```

Now UNPurge the volume on disc number 1.

```
ESC,c UNP V on D#1;
or
ESC,c UNPurge Volume on Disc#1 ;
```

```
Local   Disc       Start of    Space  Sector   Create        Create    Write
Volume  Type#Unit  Free Space  Left   Stagger  Date (M/D/Y)  Time      Protect
------  ---------  ----------  -----  -------  ------------  --------  -------
DRAIN    mfD#1            37   1019       9      10/21/81    12:54 PM    NO
```

VERify Volume. The VERify Volume command sequence reads the entire volume specified in the command, while checking CRC's and recording tolerances. The operation takes several minutes to execute. The message "Disc VERIFY in progress" is displayed in the message window during the execution.

Example: Verify the volume name "GARY".

```
ESC,c VERIFY Volume GARY ;
or
ESC,cVER V GARY;
```

Using The ESC&p Generalized Escape Sequence

The terminal's display and external printer can be controlled by a program executing in a host computer through use of an escape sequence of the following general form:

```
ESC&p <commands>
```

where each command consists of an optional numeric parameter followed by one of the command characters shown in Table 5-4. The final command in the sequence is identified by an uppercase command character; all prior commands in the sequence use lowercase command characters. The characters "ESC&p" must not be separated by intervening blanks; blanks occuring subsequently in the escape sequence are ignored.

Table 5-4. Device Control Command Characters

Command Character		Meaning
Upper-case	Lower-case	
S	s	Defines the source ("from") device.
D	d	Defines the destination ("to") device.
C	c	Specifies the desired CTU emulator or external printer control operations.
U	u	Specifies on which terminal device (left CTU, right CTU, or printer) the specified control operation is to be performed.
P	p	For tape spacing operations, specifies the number of records or files to be spaced over. For external printer "skip lines" operations, specifies the number of lines to be skipped over.
~	^	Reads device status (see Section 10).
R	r	Reads a record or file from the current source device.
W	w	Writes a record to the current destination device(s).
B	b	Using the current source and destination device(s), copies or compares one record.
F	f	Using the current source and destination device(s), copies or compares one file.
M	m	Using the current source and destination device(s), copies or compares to the end of medium (copy all or compare all).

NOTE: The escape sequences ESC&pB, ESC&pF, and ESC&pM will not be executed when incorporated as part of a file which is to be executed using the READ key.

Within each device control escape sequence, you may specify only one physical operation (such as rewinding, spacing, reading, writing, copying, or comparing). For example, if you wish to programmatically rewind the left tape (the set of CTU emulator files assigned to CTUL), rewind the right tape, and then copy a file from one tape to the other, you would have to issue three separate escape sequences. You may, however, include sources (s), unit (u), number of records/files (p), and multiple destination (d) specification commands within an escape sequence containing one of the physical operation commands.

You should not initiate a device control operation (other than a status request) before a previous operation has been completed. For example, after initiating a read operation, the data record must be read by your program before initiating another device control operation; otherwise the read operation may not be executed properly.

Some of the device control operations can be interrupted and prematurely terminated by pressing the RETURN key. If a control operation is userinterruptable, it has the following general characteristics:

- Once the operation is initiated the terminal's keyboard is disabled except for the RETURN key.
- Pressing the RETURN key terminates the operation, returns a completion code to your program indicating that a user interrupt occurred, and reenables the keyboard.
- The completion code (if any) is transmitted when the operation is completed.
- Completion of the operation is indicated by receipt of the completion code or (for read operations) by receipt of all expected data.

Non-interruptable control operations have the following general characteristics:

- The terminal's keyboard is enabled throughout the entire operation.
- The completion code (if any) is transmitted when the operation is initiated.
- Completion of the operation is indicated by the "busy" device status bit (bit 1, byte 1) being cleared to "0".

The following device control operations are always interruptable:

Locate end-of-data (3c)
Turn on write-backspace-read mode (9c)
Turn off write-backspace-read mode (10c)
Copy record (b)
Copy file (f)
Copy all (m)
Compare record, file, or all (1m)
Device-to-datacomm data transfers
Datacomm-to-device data transfers
External printer control operations

The following operations are always non-interruptable:

Rewind (0c)
Write file mark (5c)
Write end-of-data mark (6c)

The following operations are non-interruptable when initiated locally at the terminal but are interruptable when initiated remotely over the datacomm line:

Space "p" records (1c)
Space "p" files (2c)
Locate file "p" (2c)
Space "p" records immediately without writing end-of-data mark (8c)

Any errors in a device control escape sequence cause the entire sequence to be ignored by the terminal. This could cause the CPU to go into a wait loop if your program is expecting a response upon completion of the escape sequence. You can use a programmed time-out to avoid this problem.

DEVICE CONTROL COMPLETION CODES. For tape control operations, device-to-device data transfer operations, and ASCII or binary write operations, you determine whether or not the operation was performed successfully by executing an INPUT or similar instruction which requests one ASCII character from the terminal. The terminal responds by sending an S,F, or U. An "S" indicates successful completion, an "F" indicates that the operation failed, and a "U" indicates that the operation was interrupted by the terminal operator pressing the RETURN key. Note that these completion codes cannot be suppressed by strap settings or any other means. They are always transmitted and your programs should include input commands explicitly for accepting them.

For ASCII or binary read operations, successful completion is indicated by receipt of all expected data. An I/O failure or end-of-file is indicated by receipt of just a CR(LF),RS, or GS code (depending upon the protocol employed). Datacomm errors are reported by way of the Primary Terminal Status Bytes (see Section 10).

Once your program issues a device control escape sequence to the terminal, the terminal queues further data received from the host computer in a buffer. The queued data is not acted upon until the terminal has transmitted the completion code (if any) for the current device control operation back to the host computer.

SELECTING "from" AND "to" DEVICES. To select source ("from") and destination ("to") devices, use the following commands in a device control escape sequence:

```
<"from" device code> s
<"to" device code> d
```

where the device code are as follows:

1=left tape unit
2=right tape unit
3=display
4=external printer
5=HP-IB device

Note that if the "s" or "d" command character is not preceded by a numeric code, the existing "from" or "to" device assignment (respectively) remains in effect.

The following escape sequence selects the right tape unit as the "from" device and the external printer as the "to" device:

```
ESC&p2s4D
```

Within a single device control escape sequence, you may select only one source device but you may select multiple destination devices. For example, the following escape sequence selects the left tape as the source device and both the display and the right tape as destination devices:

```
ESC&p1s2d3D
```

Once you select source and destination devices, the designated device assignments remain in effect until:

1. They are changed by a subsequent device control escape sequence;

2. They are changed by a keyboard entry;

3. The terminal operator manually initiates a "hard" reset or a program executing in the host computer initiates a "hard" reset by transmitting an ESC E sequence.

4. The terminal's power is turned off and then back on.

After a hard reset or after the terminal's power is first turned on, the "to" and "from" device assignments are as follows:

Source ("from") device = left tape
Destination ("to") device = right tape

TAPE CONTROL OPERATIONS. Although the terminal contains no facilities for tapes, means are provided, through the Cartridge TapelEmulation facility, for executing some tape commands as though tape facilities existed in the terminal. The commands are applied to "CTU emulation" files, on disc. Not all tape commands are implemented and some which are implemented are implemented only partially. For further explanation of CTU emulation, refer to Cartridge Tape Emulation in Section 6.

To perform tape control operations (such as rewinding, record spacing and locating files), use the c, u, and p commands in a device control escape sequence.

The "c" command specifies which physical operation is to be performed, as follows:

Control Code	Operation	Default Device
0	Rewind	"from"
1	Space "p" records	"from"
2	Space "p" files or locate file "p"	"from"
3	Locate end-of-data mark	"from"
5	Write file mark	"to"
6	Write end-of-data mark	"to"
8	Space "p" records immediately without writing end-of-data mark	"to"
9	Turn on write-backspace-read mode	(none)
0	Turn off write-backspace-read mode	(none)

If no control code precedes the "c", code zero (rewind) is the default.

The "u" command specifies on which tape unit the specified operation is to be performed, as follows:

1 = left
2 = right

If the "u" command is omitted or if the "u" is not preceded by a "1" or "2", the current "from" or "to" device is used (as designated above). Control codes "9" and "10" always apply to both the left and right tape unit regardless of what is specified by "s", "d", or "u" commands.

For record spacing operations (1c or 1C), the "p" command specifies the number of records to be spaced over. A positive integer preceding the "p" specifies forward spacing while a negative integer specifies backspacing.

For file spacing operations (2c or 2C), the "p" command specifies either the number of files to be spaced over or the number of the file to be located. A positive integer preceding the "p" specifies forward spacing, a negative integer specifies backspacing, and an unsigned integer specifies the number of the file to be located.

If the "p" command is omitted or if no integer precedes the "p", the value +1 is used by default.

Rewind (0c). The following escape sequence rewinds the left tape:

```
ESC&p1u0C
```

The following escape sequence rewinds the right tape:

```
Esc&p2u0C
```

Space "p" Records (1c). The following escape sequence spaces the right tape forward 12 records:

```
Esc&p+12p2u1C
```

The following escape sequence spaces the left tape backward four records:

```
Esc&p-4p1u1C
```

If a file mark is the last record encountered while backspacing, the tape is automatically spaced forward so that it is positioned just before the first record of the file. In addition, the end-of-file mark status bit (bit 4 of tape status byte 0) is set. When the tape is located at the start of a file, to reposition it just before the preceding tape mark you must backspace two records (the tape mark being one of the records) and then forwardspace one record.

Locate File "p" (2c). The following escape sequence locates the start of the third file on the left tape:

```
Esc&p3p1u2C
```

The following escape sequence locates the start of the fifth file on the right tape:

```
Esc&p5p2u2C
```

You may use either 0p or 1p to locate the load point on a tape.

Locate End-of-Data (3c). The following escape sequence locates the end-of-data mark on the left tape.

```
Esc&p1u3C
```

The following escape sequence locates the end-of-data mark on the right tape:

```
Esc&p2u3C
```

After executing either of the above escape sequences, the tape on the selected unit is positioned at the end-of-data mark. You do NOT need to reposition (backspace) the tape to add data to it. If you copy data to a tape that is positioned at the end-of-data mark, the data is appended to the last file on the tape. To add a new file to a tape that is positioned at the end-of-data mark, you must first write a file mark and then copy thhe data to the tape.

Write File Mark (5c). The following escape sequence writes a file mark on the left tape:

```
Esc&p1u5C
```

The following escape sequence writes a file mark on the right tape:

```
Esc&p2u5C
```

Write End-of-Data Mark (6c). The following escape sequence writes an end-of-data mark on the left tape:

```
Esc&p1u6C
```

The following escape sequence writes an end-of-data mark on the right tape:

```
Esc&p2u6C
```

Turn On Write-Backspace-Read Mode (9c). The HP 2640 series terminals includeqa write-backspace-read capability which provides automatic verification of data written to tape. When this capability is turned on, each time a record is written to tape the tape is automatically backspaced and the record is read and compared against the original data. If a discrepancy is detected, the terminal tries to write the record again. If the record cannot be successfully written in nine retries, both the write error (bit 1, byte 0) and hard error (bit 3, byte 2) tape status bits are set. If the record is successfully written during one of the retries, the soft error (bit 4, byte 2) tape status byte is set.

The following escape sequence turns on the write-backspace-read mode.

```
Esc&p9C
```

Turn Off Write-Backspace-Read Mode (10c). The following escape sequence turns off the write-backspace-read mode:

```
Esc&p10C
```

DEVICE-TO-DEVICE OPERATIONS. You can use device control escape sequences to copy or compare data between terminal devices.

Copy Record (b). To copy a record from a terminal device to one or more other terminal devices, use the command character "b" in a device control escape

sequence. Within the same escape sequence you may also specify the source and destination devices by using the "s" and "d" command characters in conjunction with the appropriate parameter values. If you omit a source and destination device specification, the current "from" and "to" device assignments are used.

Some examples are as follows:

`&p3s2dB`	Copy one record (the line containing the cursor) from the display to the right tape.
`&pB`	Copy one record from the current "from" device to the current "to" device(s).
`&p1s2d3dB`	Copy one record from the left tape to both the display and the right tape.
`&p1sB`	Copy one record from the left tape to the current "to" device(s).
`&p3dB`	Copy one record from the current "from" device to the display.
`&p3s4dB`	Copy one record (the line containing the cursor) from the display to the external printer.

Any file or end-of-data marks on the "from" device are copied to the "to" devices and count as one record each. No file marks are transferred, however, when the display is the "from" device and format mode is off.

Both of the following are considered to be errors:

1. The "from" device is located at the end-of-data when the copy operation is initiated.
2. An attempt is made to copy a record beyond the available data space of a "to" device.

Copy File (f). To copy a file from a terminal device to one or more other terminal devices, use the command character "f" in a device control escape sequence. Within the same escape sequence you may also specify the source and destination devices by using the "s" and "d" command characters in conjunction with the appropriate parameter values. If you omit a source and destination device specification, the current "from" and "to" device assignments are used.

When a non-formatted display is the source device, the copy file operation starts at the first character position of the line containing the cursor and copies through the last character position currently displayed on the screen. No file mark is copied.

When a formatted display is the source device, the copy file operation starts at the current cursor position and copies all "unprotected" and "transmit only" fields through the end of display memory, at which time a file mark is also copied.

When a CTU Emulator file is the source device, the copy file operation starts at the current tape position and copies until either a file mark or an end-of-data mark is detected. Upon completion, the mark itself is copied.

When a non-formatted display is the destination device, the copy file operation starts at the current cursor position. File marks copied from the source device are discarded.

When a formatted display is the destination device, the copy file operation starts at the current cursor position. Data is copied only into "unprotected" and "transmit only" fields. File marks copied from the source device are discarded.

When an external printer is the destination device, any file marks or end-of-data marks copied from the source device cause a form feed to occur.

Some examples are as follows:

`&p3s2dF`	Copy a file from the display to the right tape.
`&pF`	Copy a file from the current "from" device to the current "to" device(s).
`&p1s2d3dF`	Copy a file from the left tape to both the display and the right tape.
`&p1sF`	Copy a file from the left tape to the current "to" device(s).
`&p3dF`	Copy a file from the current "from" device to the display.
`&p3s4dF`	Copy a file from the display to the external printer.

Both of the following are considered to be errors:

1. The "from" device is located at the end-of-data when the copy operation is initiated.
2. An attempt is made to copy beyond the available data space of a "to" device.

Copy All (m). To copy all from a terminal device to one or more other terminal devices, use the command character "m" in a device control escape sequence. Within the same escape sequence you may also specify the source and destination devices by using the "s" and "d" command characters in conjunction with the appropriate parameter values. If you omit a source and destination device specification, the current "from" and "to" device assignments are used.

The copy all operation starts at the current cursor or tape position of the "from" device and copies everything through the end of medium (end of display memory or end-of-data mark).

Some examples are as follows:

`Ec&p3s2dM`	Copy all (all characters from the current cursor position through the end of display memory) from the display to the right tape.
`Ec&pM`	Copy all from the current "from" device to the current "to" device(s).
`Ec&p1s2d3dM`	Copy all (all records from the current tape position through the end-of-data mark) from the left tape to both the display and the right tape.
`Ec&p1sM`	Copy all from the left tape to the current "to" device(s).
`Ec&p3dM`	Copy all from the current "from" device to the display.
`Ec&p3s4dM`	Copy all (all characters from the current cursor position through the end of display memory) from the display to the external printer.

Both of the following are considered to be errors:

1. The "from" device is located at the end-of-data when the copy operation is initiated.
2. An attempt is made to copy beyond the available data space of a "to" device.

Compare Record (1b). To compare a record between one terminal device and another, use the command character "b" preceded by a "1" in a device control sequence. Within the same escape sequence you may also specify the source and destination devices by using the "s" and "d" command characters in conjunction with the appropriate parameter values. If you omit a source and destination device specification, the current "from" and "to" device assignments are used. Note that, unlike the copy record function, the compare record function requires that there be only one destination device.

Compare File(1f). To compare a file between one terminal and another, use the command character "f" preceded by a "1" in a device control sequence. Within the same sequence, you may also specify the source and destination devices by using the "s" and "d" command characters in conjunction with the appropriate parameter values. If you omit a source and destination device specification, the current "from" and "to" assignments are used. Note that, unlike the Copy File function, the Compare File function requires that there be only one destination device.

Some examples are as follows:

`Ec&ps2d1F`	Compare the current file on the left tape (CTUL) with the current file on the right tape (CTUR).
`Ec&p3s1d1F`	Compare the file on the display with the current file on the left tape.
`Ec&p2s3d1F`	Compare the current file on the right tape with the file on the display.
`Ec&p1d1F`	Compare the current file on the current "from" device with the current file on the left tape.
`Ec&p2s1F`	Compare the current file on the right tape with the current file on the current "to" device.
`Ec&p1F`	Compare the current file on the current "from" device with the current file on the current "to" device.

The two files are compared record-by-record. If the two records do not exactly match, the compare operation fails and is terminated. The "from" and "to" devices are left positioned immmediately following the records that do not match.

Compare All (1m). To compare all data between one terminal device and another, use the command character "m" preceded by a "1" in a device control escape sequence. Within the same escape sequence, you may also specify the source and destination devices by using the "s" and "d" command characters in conjunction with the appropriate parameter values. If you omit a source and destination device specification, the current "from" and "to" device assignments are used. Note that, unlike the Copy All function, the Compare All function requires that there be only one destination device.

Some examples are as follows:

^t&p3s2d1M	Compare all on the display (all characters from the current cursor psition through the end of display memory) with all on the right tape (all records from the current tape position through the end-of-data mark).
^t&p1s2d1M	Compare all on the left tape with all on the right tape. The compare operation starts at the current tape position (for both tapes) and proceeds through the end-of-data mark.
^t&p1d1M	Compare all on the current "from" device with all on the left tape (all records from the current tape position through the end-of-data mark).
^t&p1M	Compare all on the current "from" device with all on the current "to" device.
^t&p2s1M	Compare all on the right tape (all records from the current tape position through the end-of-data mark) with all on the current "to" device.

The data on the two devices is compared record-by-record. If the two records do not exactly match, the compare operation fails and is terminated. The "from" and "to" devices are left positioned immediately following the records that do not match.

Some examples are as follows:

^t&p1s2d1B	Compare the current record on the left tape with the current record on the right tape.
^t&p3s1d1B	Compare the line containing the cursor on the display with the current record on the left tape.
^t&p2s3d1B	Compare the current record on the right tape with the line containing the cursor on the display.

If the current record on the "to" device does not exactly match the current record on the "from" device, the compare operation fails and the "from" and "to" devices are left positioned immediately following the records that do not match.

Compare File (1f). To compare a file between one terminal device and another, use the command character "f" preceded by a "1" in a device control sequence. Within the same escape sequence you may also specify the source and destination devices by using the "s" and "d" command characters in conjunction with the appropriate parameter values. If you omit a source and destination device specification, the current "from" and "to" device assignments are used. Note that, unlike the copy file function, the compare file function requires that there by only one destination device.

See the description of the "Copy File (f)" function earlier in this section for the definition of what constitutes the current file on each possible "from" and "to" device.

Some examples are as follows:

^t&p1s2d1F	Compare the current file on the left tape with the current file on the right tape.
^t&p3s1d1F	Compare the current file on the display with the current file on the left tape.
^t&p2s3d1F	Compare the current file on the right tape with the current file on the display.
^t&p1d1F	Compare the current file on the current "from" device with the current file on the left tape.
^t&p2s1F	Compare the current file on the right tape with the current file on the current "to" device.
^t&p1F	Compare the current file on the current "from" device with the current file on the current "to" device.

The two files are compared record-by-record. If the records do not exactly match, the compare operation fails and is terminated and the "from" and "to" devices are left positioned immediately following the records that do not match.

Compare All (1m). To compare all data between one terminal device and another, use the command character "m" preceded by a "1" in a device control escape sequence. Within the same escape sequence you may also specify the source and destination devices by using the "s" and "d" command characters in conjunction with the appropriate parameter value. If you omit a source and destination device specification, the current "from" and "to" device assignments are used. Note that, unlike the copy all function, the compare all function requires that there be only one destination device.

Some examples are as follows:

^t&p3s2d1M	Compare all on the display (all characters from the current cursor position through the end of display memory) with all on the right tape (all records from the current tape position through the end-of-data mark).

ESC&p1s2d1M — Compare all on the left tape with all on the right tape. The compare operation starts at the current tape position (for both tapes) and proceeds through the end-of-data mark.

ESC&p1d1M — Compare all on the current "from" device with all on the left tape (all records from the current tape position through the end-of-data mark).

ESC&p1M — Compare all on the current "from" device with all on the current "to" device.

ESC&p2s1M — Compare all on the right tape (all records from the current tape position through the end-of-data mark) with all on the current "to" device.

The data on the two devices are compared record-by-record. If the records do not exactly match, the compare operation fails and is terminated and the "from" and "to" devices are left positioned immediately following the records that do not match.

EXTERNAL PRINTER CONTROL OPERATIONS. To cause line feeds or form feeds on an external printer, use the c, u, and p commands in a device control sequence.

When directing a device control escape sequence to an external printer, you should explicitly use the command string "4u" or "4U" in the escape sequence.

The "c" command specifies which physical operation is to be performed. When the escape sequence is directed to an external printer, the "c" control codes have the following meanings:

Control Code	Operation
0	Form feed
1	Skip "p" lines
2–10	Form feed

For the skip lines function (1c or 1C), the "p" command specifies how many lines are to be skipped. The terminal uses the absolute value of the integer immediately preceding the "p" and sends that many ASCII line feed (LF) control codes to the external printer.

Some examples are as follows:

ESC&p0c4U — Initiate a form feed on the external printer.

ESC&p1c4u6P — Initiate six consecutive line feeds on the external printer.

ESC&p4u8C — Initiate a form feed on the external printer.

TERMINOLOGY. The following terminology is used subsequently in this chapter to describe the format of the data being passed between terminal devices and the datacomm link.

Record

A line of text from the display or a physical record on tape. Records from the display, and records copied from the display to tape, are terminated by a CR LF which are counted as two bytes. The only exception is display records which contain an explicit LF in any column other than column zero: in such a case no ESC is added and the LF designates the end of the record. A record may contain from one to 256 bytes of data (there are no null records).

A record from a formatted display (possibly copied to tape) contains one or more fields, each field copied from a single unprotected or transmit-only field in the form. Such data should be read using an ASCII read operation, in which case each field is treated as a unit of data.

File

On an unformatted display, a file consists of all records starting with the one containing the cursor and continuing through the end of the data currently displayed on the screen (zero to 24 records). On a formatted display, a file consists of all "unprotected" and "transmit only" data from the current cursor position through the end of display memory. In both cases, a block terminator (either an ASCII ETB or ETX, as described below) or a non-displaying terminator is considered the end of display memory.

For CTU emulator files, a file consists of all records from the current tape position up to and including the next file mark. The number of records within a tape file is limited only by the capacity of the file.

Handshake

In a point-to-point configuration, the type of handshake protocol used depends upon how switches D, G, and H on the Keyboard Interface PCA are set.

If switches G and H are both open, no handshaking occurs.

If switch G is closed, the handshake merely consists of an ASCII ENQ transmitted from the computer to the terminal.

If switch G is open and switch H is closed, the handshaking protocol is as follows:

1. Computer sends ENQ.
2. Terminal responds with ACK (followed by CR [LF] if switch D is closed. The LF is included in the AUTO LF switch on the keyboard is latched down).
3. Computer responds with another ENQ.

End of Text <end>
In a point-to-point, half duplex configuration, the line is turned around. If main channel protocol is being used, an end-of-text character is sent by the terminal (an ASCII ^X or ^T character, depending upon the setting of switch T on the terminal's Keyboard Interface PCA).

In a point-to-point, full duplex configuration, nothing happens as the result of an end-of-text condition.

Block Terminator <^S or ^S>
In a point-to-point configuration, the block terminator is an ASCII ^S.

DEVICE-TO-DATACOMM DATA TRANSFERS.

You can use device control escape sequences to initiate a "read operation" that transfers data from a terminal device (the display or the left or right tape) to your program over a data communications link.

ASCII (7-Bit) Read Operations. To initiate an ASCII (7-bit) read operation, use the following device control escape sequence:

```
ESC&p [<device code> s] <read control byte> R
```

where:

`device code`	is an optional parameter which, if present, specifies the "from" device as follows: 1 = left tape 2 = right tape 3 = display 5 = HP-IB device
`read control byte`	specifies the type of read operation, as follows: 0 = ASCII, read a record 1 = ASCII, repeat last record 4 = ASCII, read a file

This escape sequence is recognized as valid only when received over the datacomm line. It is ignored if issued locally at the terminal.

The source ("from") device may be selected by way of a separate device control sequence. If the source device specification is omitted from the ASCII read device control escape sequence, the current source device is used.

The ASCII read device control escape sequence merely tells the terminal's maincode what to do in response to the next handshake; the read operation itself is initiated by the handshake. With the HP 3000 Computer System, for example, the handshake is automatically initiated by the device driver within the operating system in response to the next input command executed in your computer program.

For an ASCII (7-bit) read operation, data is sent to the datacomm as seven-bit ASCII characters. Eight-bit internal display codes (such as "start unprotected field" or "start inverse video") are translated into their equivalent escape sequences (such as ESC[or ESC&dB). Unrecognized eight-bit codes are translated into DEL codes. If a device record ends with ^M^J, the ^M^J is stripped prior to transmission of the record (^M and ^J codes are, however, added as shown later in this topic). In addition any ^J codes within a record are stripped. Parity is added according to the settings of the keyboard parity switch and switch Z or the terminal's keyboard interface PCA.

The "repeat last record" operation (read control byte = 1) is only valid following a "read record" or "repeat last record" operation (read control byte = 0 or 1). In such a case the previously-transmitted record or field is retransmitted. Any intervening data transfer, such as a local copy between devices or an escape sequence write operation, renders the "repeat last record" request invalid.

The other ASCII read operations (read control byte = 0 or 4) are described below. In each case the escape sequence is issued explicitly (via a PRINT or equivalent output statement) by the program executing in the host computer. The handshake and end of text sequences (denoted by <handshake> and <end>, respectively) are as described under "Terminology" earlier in this section. The optional ^J code is transmitted by the terminal if the AUTO LF key on the keyboard is latched down.

Character or Block Line Mode, Read Normal Record:

Computer	Terminal
ESC&p0R	
<handshake>	
	<record> ^M[^J] <end>

Character or Block Line Mode, Read Format Mode Record:

Computer	Terminal
ESC&p0R	
<handshake>	
	<field> ^M[^J] <end>

Character or Block Line Mode, Read End-of-File Record:

Computer	Terminal
ESC&p0R	
<handshake>	
	<^S or ^S> ^M[^J] <end>

Character or Block Line Mode, Read Normal File:

```
Computer                              Terminal
ESC&p4R
        <handshake>
                                <record 1> %[LF] <end>
        <handshake>
                                <record 2> %[LF] <end>
                                        .
                                        .
                                        .
        <handshake>
                                <record n> %[LF] <end>
        <handshake>
                                  <% or %> %[LF] <end>
```

Character or Block Line Mode, Read Format-Mode File:

```
Computer                              Terminal
ESC&p4R
        <handshake>
                                <field 1> %[LF] <end>
        <handshake>
                                <field 2> %[LF] <end>
                                        .
                                        .
                                        .
        <handshake>
                                <field n> %[LF] <end>
        <handshake>
                                  <% or %> %[LF] <end>
```

Block Page Mode, Read Normal Record:

```
Computer                              Terminal
ESC&p0R
        <handshake>
                          <record> %[LF] <% or %> <end>
```

Block Page Mode, Read Format-Mode Record:

```
Computer                              Terminal
ESC&p0R
        <handshake>
                           <field> %[LF] <% or %> <end>
```

Block Page Mode, Read End-of-File Record:

```
Computer                              Terminal
ESC&p0R
        <handshake>
                                      <% or %> <end>
```

Block Page Mode, Read Normal File:

```
Computer                              Terminal
ESC&p4R
        <handshake>
                          <record 1> %[LF] <% or %>
                          <record 2> %[LF] <% or %>
                                  .
                                  .
                                  .
                          <record n> %[LF] <% or %>
                                    <% or %> <end>
```

Block Page Mode, Read Format-Mode File:

```
Computer                              Terminal
ESC&p4R
        <handshake>
                          <field 1> %[LF] <% or %>
                          <field 2> %[LF] <% or %>
                                  .
                                  .
                                  .
                          <field n> %[LF] <% or %>
                                    <% or %> <end>
```

Binary (8-Bit Read Operations. To initiate a binary (8-bit) read operation, use the following device control escape sequence:

```
ESC&p [<device code> s] <read control byte> R
```

where:

`device code` is an optional parameter which, if present, specifies the "from" device as follows:

1 = left tape
2 = right tape
3 = display
5 = HP-IB device

`read control byte` specifies the type of read operation, as follows:

2 = binary, read a record
3 = binary, repeat last record
6 = binary, read a file

This escape sequence is recognized as valid only when received over the datacomm line. It is ignored if issued locally at the terminal.

The source ("from") device may be selected by way of a separate device control sequence. If the source device specification is omitted from the binary read device control escape sequence, the most recently specified source device is used.

The binary read device control escape sequence merely tells the terminal's maincode what to do in response to the next handshake; the read operation itself is initiated by the handshake. With the HP 3000 Computer System, for example, the handshake is automatically initiated by the device driver within the operating system in response to the next input command executed in your computer program.

For a binary (8-bit) read operation, data is sent to the datacomm without interpretation. No characters are added or deleted. The eighth (high-order) bit of each byte may or may not be replaced with parity, depending upon the factors detailed below.

A byte count is sent prior to transmission of each record. The byte count consists of four bytes of mapped hexadecimal, using the following mapping (the most significant byte is sent first):

```
Normal Hex: 0 1 2 3 4 5 6 7 8 9 A B C D E F
264x:       0 1 2 3 4 5 6 7 8 9 : ; < = > ?
```

For example, if the record contains sixty bytes (3C hex), the terminal transmits a byte count of 003<.

For point-to-point configurations, if 8-bit data is desired (i.e., no parity), the keyboard parity switch must be set to "NONE" and switch Z on the terminal's keyboard interface PCA must be closed to defeat parity generation.

The "repeat last record" operation (read control byte = 3) is only valid following a "read control" or "repeat last record" operation (read control byte = 2 or 3). In such a case the previously-transmitted record is retransmitted. Any intervening data transfer, such as a local copy between devices or an escape sequence write operation, renders the "repeat last record" request invalid.

The other binary read operations (read control byte = 2 or 6) are described below. In each case the escape sequence is issued explicitly (via a PRINT or equivalent output statement) by the program executing in the host computer. The handshake and end of text sequences (denoted by <handshake> and <end>, respectively) are as described under "Terminology" earlier in this section. The optional ␍␊ code is transmitted by the terminal if the AUTO LF key on the keyboard is latched down.

Character or Block Line Mode, Read Normal or Format-Mode Record:

```
Computer                                  Terminal
ESC&p2R
        <handshake>
                                  <count> %[␍␊] <end>
        <handshake>
                                      <record> <end>
```

Character or Block Line Mode, Error or Read End-of-File Record:

```
Computer                                  Terminal
ESC&p2R
        <handshake>
                                <% or %> %[␍␊] <end>
```

Character or Block Line Mode, Read File:

```
Computer                                  Terminal
ESC&p6R
        <handshake>
                                  <count 1> %[␍␊] <end>
        <handshake>
                                      <record 1> <end>
                                              .
                                              .
                                              .
        <handshake>
                                  <count n> %[␍␊] <end>
        <handshake>
                                      <record n> <end>
        <handshake>
                                <% or %> %[␍␊] <end>
```

Block Page Mode, Read Normal or Format-Mode Record:

```
Computer                                  Terminal
ESC&p2R
        <handshake>
                                <count> <% or %> <end>
        <handshake>
                                      <record> <end>
```

Block Page Mode, Error or Read End-of-File Record:

```
Computer                                  Terminal
ESC&p2R
        <handshake>
                                      <% or %> <end>
```

Block Page Mode, Read File:

```
Computer                                  Terminal
ESC&p6R
        <handshake>
                  <count 1> <% or %> <record 1>
                  <count 2> <% or %> <record 2>
                                      .
                                      .
                                      .
                  <count n> <% or %> <record n>
                                <% or %> <end>
```

Fast Binary (Program Load) Read Operations. To pass binary data from a terminal device directly into a host computer without any handshake protocol, use the following escape sequence:

```
ESCe
```

This escape sequence may be issued either locally or from a program executing in the host computer. In response to this escape sequence, a file from the current source ("from") device is copied to the datacomm without interpretation.

The entire file is transmitted without byte counts, delimiters, or any other characters inserted into the data stream. There is no handshake. At the end of the transmission, two bytes of all zeros (000 octal) are sent if there were no errors: otherwise two bytes of all ones (377 octal) are sent.

For point-to-point configurations, parity generation is automatically suppressed. If switch F on the terminal's keyboard interface PCA is closed, the transmission occurs at the baud rate specified by the keyboard BAUD RATE switch. If switch F is open, the transmission rate is 9600 baud.

Two escape sequences are provided by programmatically initiating Record mode, sending an end-of-data signal to the terminal, and terminating Record mode, the "Copy File from DAtacomm", or "Copy All from DAtacomm" state. The escape sequences are used to define an end-of-data code which, when received as the first character of a record in the data transmission, is interpreted as the end of data.

The escape sequences are as follows:

```
ESC&p <code>p 21C
```

where:

<code> = 1 thru 127	The corresponding ASCII code character is defined as the end-of-data character when it appears at the start of a record in a Record, "Copy File from DAtacomm", or "Copy All from DAtacomm" operation.
<code> = 0 or none	The end-of-data feature is disabled.

NOTE: This sequence always returns an "S" (successful) indication.

```
ESC&p <code>p 20C
```

where:

<code> = 1 thru 127	Record mode is initiated and the ASCII code character corresponding to "code" is defined as the end-of-data character. When Record mode is ended, the previous end-of-data character becomes active.
<code> = 0	The end-of-data feature is disabled for this operation only.
<code> = none	The currently-selected end-of-data character (if any) is used.

NOTES:

1. This sequence returns an "S" indication only if the Record operation was successful.

2. A DC1 (completion code) should be transmitted following the end-of-data character.

DATACOMM-TO-DEVICE DATA TRANSFERS. You can use device control escape sequences to initiate a "write" operation that transfers data from your program to a terminal device (the display or the left or right tape) over a data communications link.

ASCII (7-Bit) Write Operations. To initiate an ASCII (7-bit) write operation, use the following device control escape sequence:

```
ESC&p [<device code> d] W <record>
```

where:

device code	is an optional parameter which, if present, specifies the "to" device as follows: 1 = left tape 2 = right tape 3 = display 4 = external printer 5 = HP-IB device
device	is the data to be transmitted.

This escape sequence is recognized as valid only when received over the datacomm line. It is ignored if issued locally at the terminal.

The record being transmitted is terminated by the 256th data byte after the "W" or by the first ↳ code, whichever comes first. On some terminals, one-byte records (containing a single ↳ code) cannot successfully be written to the display.

The data is sent from the datacomm to one or more terminal devices as seven-bit characters. The eighth (high-order) bit may be checkked for parity, depending upon the settings of the keyboard PARITY switch, and switch Z on the terminal's keyboard interface PCA. The eighth bit is then cleared.

If any transmission errors occur, the entire record being transmitted is discarded and the operation aborted.

The destination ("to") device(s) may be selected by way of a separate device control sequence. If a destination device specification is omitted from the ASCII write device control escape sequence, the current destination device(s) is used.

If a handshake (initiated by an INPUT or similar command) is performed following a write escape sequence, the terminal transmits an ASCII "S" or "F" to indicate whether or not the write operation was successfully performed. For interrupt-driven devices (such as tapes), the "S" indicates that the write was succcessfully initiated. For other devices, the "S" indicates successful completion. The "F" indicates failure. If a datacomm error occurs during transmission, the completion code is unpredictable. Datacomm errors are reported by way of the terminal status bytes.

For point-to-point configurations, NUL and DEL codes are normally stripped from the incoming data. In addition, the terminal responds to an incoming ENQ code by transmitting an ACK; the ENQ code is not considered part of the incoming data.

In the following definitions, the optional LF code is transmitted by the terminal if the [AUTO LF] key on the keyboard is latched down.

Character or Block Line Mode:

Computer	Terminal
ESC&p W <record>	
<handshake>	
	<S or F> ENQ[LF]

Block Page Mode:

Computer	Terminal
ESC&p W <record>	
<handshake>	
	<S or F> <ENQ or NAK>

Binary (8-Bit) Write Operations. To initiate a binary (8-bit) write operation, use the following device control escape sequence:

```
ESC&p [<device code> d] <byte count> W <record>
```

where:

`<device code>` is an optional parameter which, if present, specifies the "to" device as follows:
1 = left tape
2 = right tape
3 = display
4 = external printer
5 = HP-IB device

`byte count` is a decimal integer within the range 1-256 which specifies the number of bytes to be transmitted. If the byte count is zero, an ASCII (7-bit) write operation is performed as described above.

`record` is the data to be transmitted.

This escape sequence is recognized as valid only when received over the datacomm line. It is ignored if issued locally at the terminal.

The record is transmitted from the datacomm to all "to" devices. The destination ("to") device(s) may be selected by way of a separate device control sequence. If a destination device specification is omitted from the binary write device control escape sequence, the current destination device(s) is used. See the discussions of the individual protocols below to determine whether seven-bit or eight-bit data is passed.

If the data is passed to the display or an external printer, bytes with the high-order bit set are interpreted by the terminal as internal display codes (such as "start unprotected field" or "start inverse video"). Transmission of display code escape sequences to an external printer is enabled by opening switch N on the terminal's keyboard interface PCA.

Data passed to the tapes is copied exactly as received.

If any transmission errors occur, the entire record being transmitted is discarded and the operation aborted.

If a handshake (associated with an INPUT or similar command) is performed following a write escape sequence, the terminal transmits an ASCII "S" or "F" to indicate whether or not the write operation was successfully performed. For interrupt-driven devices (such as tapes), the "S" indicates that the write was successfully initiated. For other devices, the "S" indicates successful completion. The "F" indicates failure. If a datacomm error occurs during transmission, the completion code is unpredictable. Datacomm errors are reported by way of the terminal status bytes.

For point-to-point configurations, you must transmit an ENQ to the terminal, and receive back an ACK, after transmitting the "W" in order to allow the terminal enough time to switch into binary mode. When the terminal is in binary mode, parity checking is automatically disabled. If the keyboard PARITY switch is set to "EVEN" or "ODD", the high-order bit of each data byte is cleared and seven-bit data is passed to the "to" device(s). If the keyboard PARITY switch is set to "NONE", all eight bits of each data byte are passed to the "to" device(s). During transmission of the record, DC1, DC3, and DEL codes are passed as data (the ACK-ENQ handshake is disabled) until the byte count has been satisfied.

In the following definitions, the optional LF code is transmitted by the terminal if the [AUTO LF] key on the keyboard is latched down.

Point-to-Point, Character or Block Line Mode:

Computer	Terminal
ESC&p <count> W ENQ	
	ACK <end>
<record>	
<handshake>	
	<S or F> ENQ[LF]
	<end>

Point-to-Point, Block Page Mode:

Computer	Terminal
ESC&p <count> W ENQ	
	ACK <end>
<record>	
<handshake>	
	<S or F> <ENQ or ETB>
	<end>

Table 5-5. Summary of ESC&p Device Control Escape Sequences

S = Source Device Assignment
D = Destination Device Assignment
P,U,C = Device Command
^ = Status
R = Read
W = Write
B = Copy (Compare) Record
F = Copy (Compare) File
M = Copy (Compare) All

```
ESC&p
[<"from" device code>s]
[<"to" device code>d]
[<control parameter>p][<device code>u]<control code>c
[<device code>]^
[<read control byte>]r
[<byte count>]w
[<control bit>]b
[<control bit>]f
[<control bit>]m
```

where:

device codes (**s**, **d**, **u**) are:

1 = left tape
2 = right tape
3 = display
4 = external printer

(If no value is specified, previous device assignments are in effect.)

control code (**c**) is:

Control Code (c)	Default Device	Function
0	"from"	Rewind
1	"from"	Space "p" records
2	"from"	Space "p" files
3	"from"	Locate end-of-data mark
5	"to"	Record file mark
6	"to"	Record end-of-data mark
8	"to"	Skip "p" records immediately without recording end-of-data mark
9		Turn on write-backspace-read mode
10		Turn off write-backspace-read mode
20		Initiates record mode and specifies that the control parameter is to be used as the end-of-data character.
21		Specifies that the control parameter is to be used as the end-of-data character. No effect on record mode.

(If no value is specified, 0 is assumed.)

control parameter (**p**) when used with control codes 1 thru 10

a positive (**+n**), negative (**-n**), or unsigned (**n**) integer, specifying the number of records or files for (**c**) control codes 1 and 2.

(If no value is specified, +1 is assumed.)

control parameter (**p** when used with control codes 20 and 21

used as the end-of-data character.

read control byte (**r**) is:

0 = transmit next record
1 = retransmit last record only
2 = send byte count before transmitting next record
3 = send byte count before retransmitting last record read
4 = transmit file
6 = transmit file with byte count before each record

(If no value is specified, 0 is assumed.)

byte count (**w**) is:

if no value is specified, ASCII is assumed. Data is received until a Line Feed (LF) character or 256 characters (maximum) are received.

if a value is specified, binary data is assumed.

control bits (**b**, **f**, **m**) are:

0 or no integer = copy record (b), file (f), or all (m) from source device to destination device.
1 = compare record (b), file (f), or all (m) on source and destination devices.

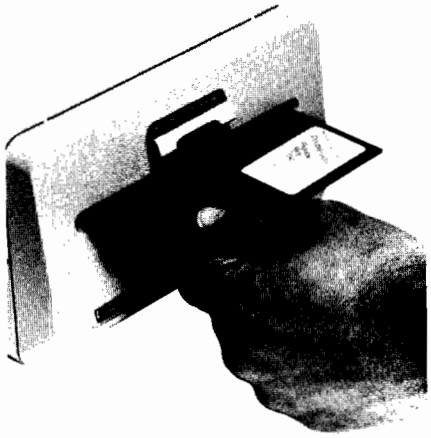
DISC DRIVE OPERATIONS FROM THE KEYBOARD

SECTION 6

This section describes how to use your terminal with the 13272A Flexible Mini Disc Drive. You should read the Commands section before reading this section to become familiar with the terminal's command channel.

THE FLEXIBLE MINI DISC DRIVE

A door in the front of the mini disc drive allows you to insert and remove the diskette. The door snaps open for inserting and removing the diskette. When inserting a diskette, make sure that there is not another diskette already in the disc drive. If there is, remove it, before inserting the other one. Be sure to close the door after inserting the diskette.



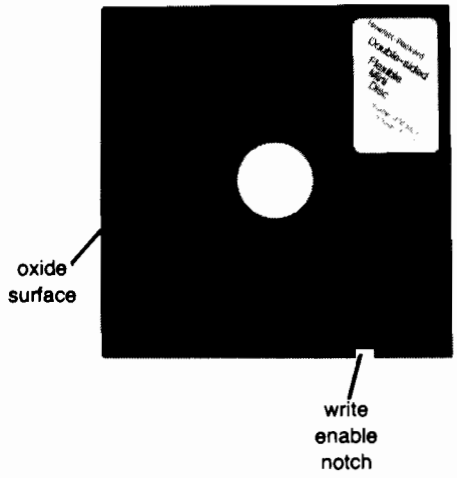
Inserting the Diskette

CAUTION

If you accidentally insert another diskette when there is one presently in the drive, BE SURE TO REMOVE THE BOTTOM ONE FIRST. Otherwise, damage to the read/write head in the drive may result.

THE DISKETTE

The diskette is similar to a phonograph record in a jacket. Slots in the jacket expose the oxide surface of the disc so that the head in the disc drive can record and read the information.



Care

- The Diskette should be kept in its envelope when not in use. This keeps dust particles and other substances from getting on the oxide surface.
- Don't bend or flex the diskette; this will warp or break the disc.
- Be sure that it is never exposed to temperatures greater than 50°C (122°F); this may warp the disc.
- Be sure that it never comes near a magnetic source; this will destroy the data on the disc and may permanently damage its recording properties.

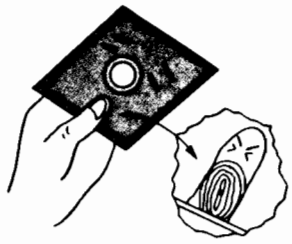
Write-Protect

A notch in the diskette jacket allows a switch in the disc drive to enable writing to the disc. If this notch is covered, then it is impossible to write to the disc; data that is presently on the disc cannot be over written. This is useful when the data on the disc should only be read, such as a form. Adhesive tabs are provided with the diskette to allow you to cover up the write/enable notch in the jacket.

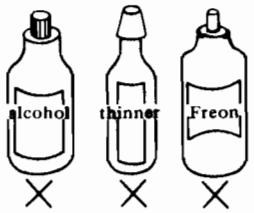
INITIALIZING A DISKETTE

Diskettes which have not been used previously are unformatted and must be formatted before they can be written to or read from. Also, the diskette (volume) requires a name.

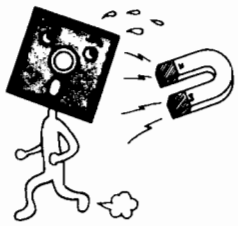
1. Do not touch the disk surface. Easily contaminated, and causes errors.



2. Do not use solutions: alcohol, thinner, Freon, to clean the disk.



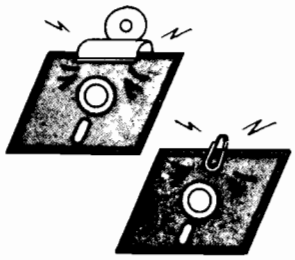
3. Do not use magnets or magnetized objects near the disk. Data can be lost from a disk when exposed to a magnetic field.



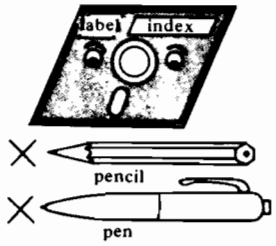
4. Do not bend or fold the disk.

5. Do not place heavy objects on the disk.

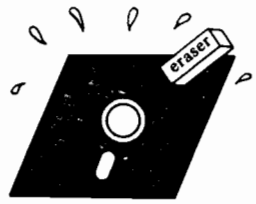
6. Do not use rubber bands or paper clips on the disk.



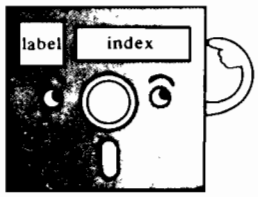
7. Do not write on a disk label with a pencil or a ball-point pen. Use a fiber-tip.



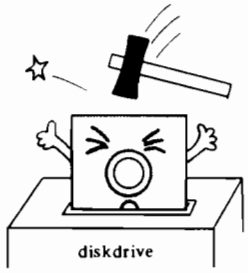
8. Do not use erasers.



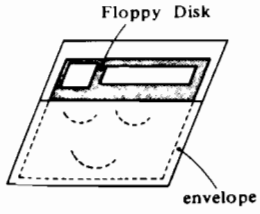
9. Put I.D. labels in a right place, never use them in layers.



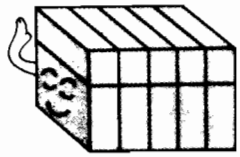
10. Insert carefully, by grasping upper edge and placing it into the drive.



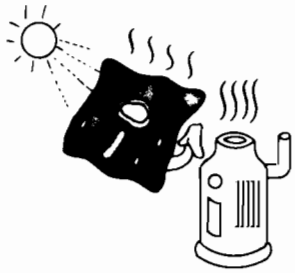
11. Keep disk in its envelope.



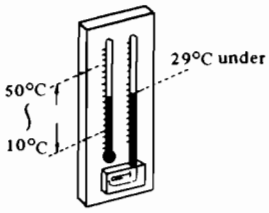
12. Store disk not for immediate use in their box, and set it upright.



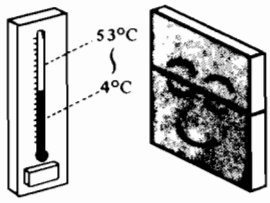
13. Do not expose the disk to excessive heat or sunlight.



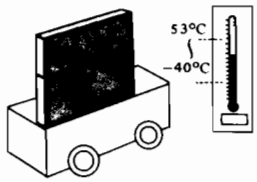
14. Operating environment
10°C to 50°C (50°F to 122°F)
20% to 80% RH
Less than 29°C (Wet bulb temperature)



15. Storage environment
4°C to 53°C (40°F to 127°F)
8% to 80% RH



16. Transportation
During transportation the disk shall be in its envelope, and in a protective box.
Temperature: −40°C to 53°C (−40°F to 127°F)
Relative humidity: 8% to 90% RH



CREATING A NEW VOLUME

The Create Volume command is used to create a new volume on a diskette. When the command is initiated, the diskette, if it has not previously been formatted, is formatted with the default stagger of 9. Then the volume name is written on the diskette and space is allocated for the file directory.

If a stagger other than the default stagger of 9 is desired, it is necessary to use the Format command to specify the desired stagger before using the Create Volume command.

Up to six characters (alpha and/or numeric) may be used to name the diskette. The first character must be alpha. If more than six characters are specified, the terminal truncates the name after six characters.

Example: Format a new diskette and assign a volume name of "MEMOS".

Step 1. Press [COMMAND] , disc , CREATE , VOLUME , MEMOS, ON , 1 [RETURN] .

Step 2. The Disc FORMAT in progress message is displayed followed by the Disc VERIFY in progress message. Each message is displayed for approximately 45 seconds.

(You may verify the new volume name by following the example in "Showing Volume and File Names" below.)

Step 3. Press [COMMAND] again to exit command mode.

If a volume has been named and formatted previously, the CREATE VOLUME not allowed message will be displayed in the message line. If the data on this volume is no longer of value, you should *purge* the volume, then enter the *Create Volume command* again.

If a volume has been *purged*, it will require naming. (It will not require formatting because it was formatted when the *Create Volume command* was first executed on the diskette.) Again, the *Create Volume command* is used in the same manner as with new diskettes.

Renaming A Volume

You can rename your diskette by the *Rename Volume command*. The renaming operation does not affect the files on the diskette; it changes only the volume name.

Example: Rename the volume "LETTER" to "MEMO".

Step 1. Press [COMMAND] , disc , RENAME , VOLUME , LETTER, to , MEMO, [RETURN] .

(You may verify the new volume name by following the example in "Showing Volume and File Names" below.)

Step 2. Press [COMMAND] again to exit command mode.

NAMING YOUR FILES

Data written on the diskette (volume), may be divided into one or more named files. You assign the *filename* when you write data to the volume, or when you create a file of specific length (see Creating Files of a Specified Length), or when you are using the Tape Emulator (see Tape Emulation). In the descriptions of command sequences that follow, you will be instructed when a filename is required.

Up to ten characters (alpha and/or numeric) may be used to name the file. The first character must be alpha. If more than ten characters are specified, the terminal truncates the name after ten characters.

CREATING FILES OF A SPECIFIC LENGTH

In most applications you will not need to create a file of a specified length. When you execute a command to write data to the diskette, the file length is handled automatically. However, in some applications you may want to create the file before writing data to it (i.e., reserve file space for future use). But when you do, you must remember that *Append* is the only command that will write data to the file. If you attempt to use another command (such as *Copy* or *Transfer*), the message File name already exists will be displayed.

The parameters that may be specified in the *Create File command* are: 1) the number of lines, 2) the number of bytes per line, and 3) file type (ASCII or Binary). (The parameters may be specified in any order.) The defaults are: one line, 80 bytes, and ASCII. If you specify more space than is remaining on the diskette, the message File size too large will be displayed. Refer to the Command section for calculating the amount of space that may be required for a file.

Example: Create a file named "TEST". (The defaults are one line, 80 bytes per line, and file type ASCII.)

Step 1. Press [COMMAND] , disc , CREATE , FILE , TEST, [RETURN] .

Step 2. Press [COMMAND] again to exit command mode.

Example: Create a file named "TEST1" with 100 lines. (The defaults are 80 bytes per line and file type ASCII.)

Step 1. Press [COMMAND] , disc , CREATE , FILE , TEST1 100, LINES , [RETURN].

Step 2. Press [COMMAND] again to exit command mode.

Example: Create a file named "TEST2" with 250 lines of 72 bytes per line. (The default is file type ASCII.)

Step 1. Press [COMMAND] , **disc** , **CREATE** , **FILE** , TEST2 100, **LINES** , 72, **BYTES** , [RETURN] .

Step 2. Press [COMMAND] again to exit command mode.

Example: Create a file named "TEST3" with 400 lines of 150 bytes per line of type Binary.

Step 1. Press [COMMAND] , **disc** , **CREATE** , **FILE** , TEST3 400, **LINES** , 150, **BYTES** , **BINARY** , [RETURN] .

Step 2. Press [COMMAND] again to exit command mode.

SHOWING VOLUME AND FILE NAMES

The volume name and file names on a diskette may be displayed by entering the *Show Volumes* or *SHow Files command.*

Example: Display the volume name of the diskette presently in the disc drive.

Step 1. Press [COMMAND] , **SHOW** , **VOLUMES** , [RETURN] .

Step 2. Press [COMMAND] again to exit command mode.

The information is presented as shown below.

```
VOLUME NAME        HP-IB ADDRESS
----------------   -------------

Local   Disc       Start of    Space  Sector   Create          Create    Write
Volume  Type#Unit  Free Space  Left   Stagger  Date (M/D/Y)    Time      Protect
------  ---------  ----------  -----  -------  ------------    ------    -------
USER    mfD#1           199      857        9       9/12/81    4:32 PM   NO
```

volume name — double-sided disc — disc drive#1 — 199 sectors have been written on — no. of sectors available — stagger assigned — date created — time created — notch is not covered: can write

```
VOLUME NAME        HP-IB ADDRESS
----------------   -------------

Local   Disc       Start of    Space  Sector   Create          Create    Write
Volume  Type#Unit  Free Space  Left   Stagger  Date (M/D/Y)    Time      Protect
------  ---------  ----------  -----  -------  ------------    ------    -------
nonfmt  mfD#1              0     1056        9       7/15/81    8:03 AM   NO
```

"nonfmt" is an unformatted diskette

```
VOLUME NAME        HP-IB ADDRESS
----------------   -------------

Local   Disc       Start of    Space  Sector   Create          Create    Write
Volume  Type#Unit  Free Space  Left   Stagger  Date (M/D/Y)    Time      Protect
------  ---------  ----------  -----  -------  ------------    ------    -------
nolabel mfD#1              0     1056        9       7/23/81   11:14 AM   NO
```

"nolabel" is a formatted, unnamed diskette

Either the *active files* or the *purged files* on a volume may be displayed.

Example: Show the active files on the volume mounted in the drive.

Step 1. Press [COMMAND] , **SHOW** , **FILES** , [RETURN] .

Step 2. Press [COMMAND] again to exit command mode.

The information is presented as shown below.

```
Volume = USER <---- (volume name)

File        File    Start    Size        Create          Create
Name        Type    Address  (sectors)   Date (M/D/Y)    Time
---------   ------  -------  ---------   ------------    --------
USERSECT2   Ascii       37         49      6/19/80       9:12 AM
FORM12      Binary      86         19      7/14/80       1:05 PM
```

file name — file type (Ascii or Binary) — starting sector address of the file — no. of sectors that the file occupies — date set by SET DATE command — time file created

Example: Show the purged files on the same volume.

Step 1. Press [COMMAND] , **SHOW** , **PURGE** , [RETURN] .

Step 2. Press [COMMAND] again to exit command mode.

The information is presented as shown below.

```
Volume = USER

File        File    Start    Size        Create          Create
Name        Type    Address  (sectors)   Date (M/D/Y)    Time
---------   ------  -------  ---------   ------------    --------
FORECAST1   Purged      32          5      5/21/80      10:48 AM
FORECAST8   Purged     128          8      7/23/80       3:20 PM
```

SELECTING FILES FOR "SOURCE" AND "DESTINATION"

The [READ] and [RECORD] keys, as well as certain command sequences, use the assigned Source and Destination files. The defaults are:

Source:	Left Tape
Destination:	Right Tape

The terminal interprets the *left tape* and *right tape* as filenames *CTULnnn* and *CTURnnn,* respectively. *nnn* is the number of the file (e.g., CTUL006).

The default source and destination assignments may be changed by entering an *Assign command.* The new assignments remain in effect until the terminal is reset or until another *Assign command* is entered.

The graphics display can be assigned as the source or destination for a data transfer: however, since graphics data is of binary type, any file created with a Copy or Tranfer command will be of type binary.

Example: Assign the Source to disc filename "JULY". (Any subsequent pressing of the [READ] key or executing a command sequence that uses the source assignment will cause the "JULY" file to be read.)

Step 1. Press [COMMAND] , ASSIGN , more , S:file , JULY, [RETURN] .

Step 2. Verify the assignment by pressing SHOW , ASSIGNS , [RETURN] .

The following listing will appear on the display:

```
USER NAME     CURRENT ASSIGNMENTS
----------    -------------------
SOURCE        JULY
DESTINATION   RIGHT TAPE
LOG           DISPLAY
PRINTER       EXTERNAL PRINTER
CTUL          CTUL???
CTUR          CTUR???
```

Step 3. Press [COMMAND] again to exit command mode.

Example: Assign Destination to disc filename "SHIPMENTS" and EXTERNAL PRINTER. (After data has been written to the file the first time, any subsequent pressing of the [RECORD] key or executing of a command sequence that uses the destination assignment will display the message File name already exists .)

Step 1. Press [COMMAND] , ASSIGN , more , D:file , SHIPMENTS, EX PRNTR , [RETURN] .

Step 2. Verify the assignment by pressing SHOW , ASSIGNS , [RETURN] .

The following listing will appear on the display:

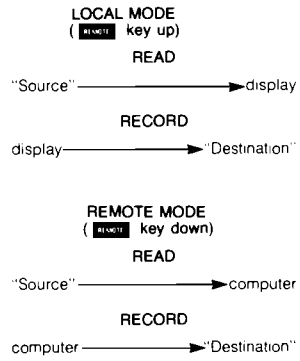
```
USER NAME     CURRENT ASSIGNMENTS
----------    -------------------
SOURCE        JULY
DESTINATION   SHIPMENT, EXTERNAL PRINTER
LOG           DISPLAY
PRINTER       EXTERNAL PRINTER
CTUL          CTUL???
CTUR          CTUR???
```

Step 3. Press [COMMAND] again to exit command mode.

RECORDING AND READING DATA (USING THE READ AND RECORD KEYS)

The [READ] and [RECORD] keys provide a convenient means to copy data to the display or datacomm from one file or device, or to copy data from the display or datacomm to a file or device. They are especially useful for repetitive operations, such as, reading files of data from a disc, or entering data on the display, then recording it on a disc file and printing it.

The [READ] and [RECORD] keys use the "Source" and "Destination" assignments and the display or datacomm, depending upon whether or not the terminal is in REMOTE ([REMOTE] key down). Here, we shall discuss only the use of these keys in Local Mode.



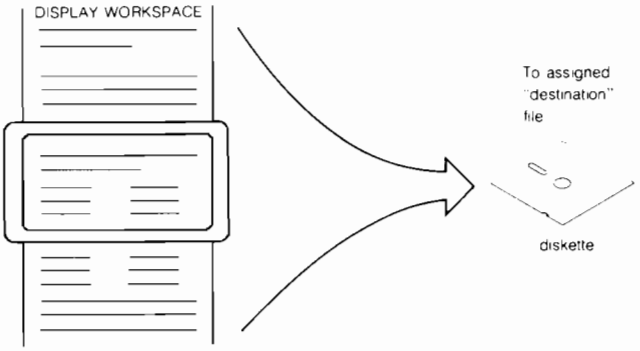
Recording Data from the Display

NOTE: The [RETURN] key may be used to stop the record operation.

Pressing [RECORD] copies data from the *display* (regardless of the "Source" file assignment) to the "Destination" file. After the [RECORD] key is pressed, the cursor is "homed" and the entire contents of the current display workspace is recorded.

There are two ways of recording data to a diskette using the [RECORD] key. First, you can define a file, by filename, using the *Assign command*, then press [RECORD] . Or, you can simply press [RECORD] without making a file assignment — tape emulation. (This assumes that the destination is the left or right tape.)

Assign "Destination" to a new filename:
Press [COMMAND] , [ASSIGN] , [more] , [D:file] , ‹filename› , [RETURN]
Press [RECORD] .



Recording Data

Non-Tape Emulator Files. In the first case, you define the "Destination" file by the *Assign command* (discussed previously), then press [RECORD] . *However, before you use the* [RECORD] *key again, the "Destination" file must be redefined.* This is because you cannot have two files with the same name on the diskette. If you attempt to record to the same file, the message **File name already exists** will be displayed in the message line.

Tape Emulator Files. In the second case, we have to assume that the default Destination is "RIGHT TAPE" or "LEFT TAPE" in the assign table. (A *Show Assignments command* will give the current assignment.) When [RECORD] is pressed, the data is written to the diskette with a filename of *CTULnnn* or *CTURnnn* (where *nnn* is the number of the file — see "CTU Emulation" at the end of this section.) A *Mark File command* must be executed after recording the data to close the file.

Example: Type some data on the screen, assign "EXERCISE" as a filename for the Destination File, and record the data on the diskette.

Step 1. Press [UP] , [CLEAR DISPLAY] , then type your name followed by [RETURN] .

Step 2. Insert a diskette into the disc drive.

Step 3. Press [COMMAND] , **ASSIGN** , **more** , **D:file** , EXERCISE, [RETURN] .

Step 4. Press [RECORD] .

Step 5. Press [COMMAND] , **SHOW** , **FILES** , [RETURN] .

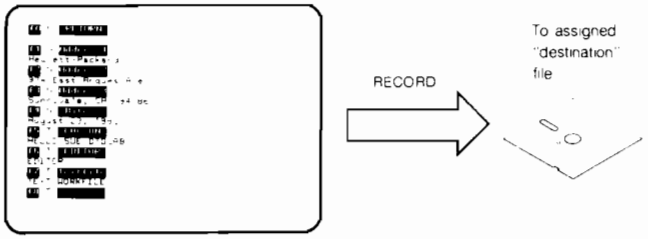
```
File       File     Start    Size       Create        Create
Name       Type     Address  (sectors)  Date (M/D/Y)  Time
---------  ------   -------  ---------  ------------  --------
CTUR001    Ascii       32        1        6/ 4/80      7:01 AM
CTUR002    Ascii       33        1        6/ 4/80     10:35 AM
EXERCISE   Ascii       34        1        6/ 4/80     11:51 AM
```

Step 6. Press [RECORD] again. Verify that the message **File name already exists** appears in the message line. As you can see, the Destination file will have to be changed to permit any further recording.

Recording the f1 — f8 Softkey Menu

The function key assignments can be recorded on the "Destination" files. Display the function key assignment menu [CNTL] [NEXT PAGE] ; press [RECORD] .

Assign "Destination" to a new filename:
Press [COMMAND] , [ASSIGN] , [more] , [D:file] , ‹filename› , [RETURN]
Press [COMMAND] again to exit command mode.
Press [CNTL] [NEXT PAGE] .
Press [RECORD] .



Recording the **f1** *through* **f8** *Softkey Assignments*

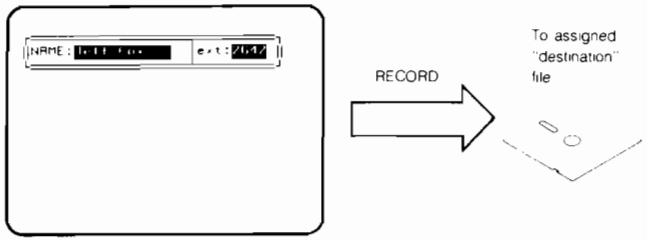
Recording Data in Format Mode

When in Format Mode (filling in a form on the display), the [RECORD] key records only the data in *unprotected fields* and the data in *transmit-only fields* of the
the [RECORD] key is pressed, the cursor is "homed" to the beginning of the first field on the form, the data is recorded on the "Destination" (a tape emulator file equivalent to the left or right tape); a file mark is recorded on the "Destination" file, then the form is cleared. This is useful when using the same form for repetitive entries.

If a printer is one of the "Destination" assignments, data in the unprotected and transmit-only fields will be printed exactly as shown on the display. This allows you to use preprinted forms.

Assign "Destination" to a new filename:
Press [COMMAND] , [ASSIGN] , [more] , [D:file] , ‹filename› , [RETURN]
Press [COMMAND] again to exit command mode.
Format Mode On: press [CNTL] [f8] or [ESC] [W]
Press [RECORD] .

NOTE When a form is to be filled in and recorded many times, either the "left tape" or the "right tape" should be selected as the destination (This technique automatically assigns the "Destination" filename each time the [RECORD] *key is pressed.)*



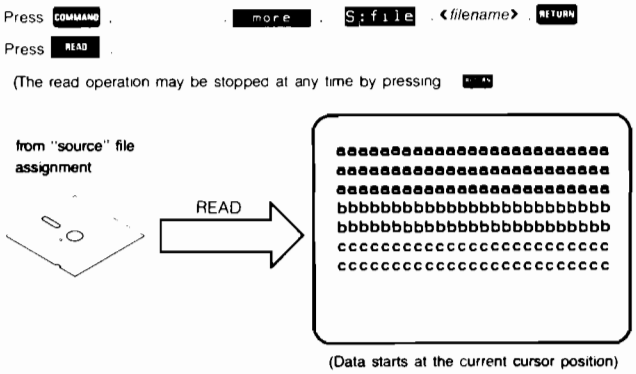
Recording Data Entered in a Form

Reading Data to the Display

Pressing READ copies data from the "Source" file assignment to the display (regardless of the "Destination" file assignment). However, the READ function differs depending upon whether or not the "Source" file is a Tape Emulator file.

Non-Tape Emulator Files. When the READ key is pressed, data is read from the beginning of the "Source" file until the RETURN key is pressed or until the end-of-file is reached. Pressing the READ key again repeats the operation (i.e., starts the read operation at the *beginning* of the file).

Tape Emulator Files. When the READ key is pressed, data is read from the *current position* of the "Source tape" until the RETURN key is pressed or until an end-of-file mark is reached. The tape may be *positioned* by "REWINDING" the tape, "FINDING" a file on the tape, or "SKIPPING LINES" on the tape. Pressing the READ key again *resumes* the read operation where the previous read left off. Consecutive files may be read by pressing READ for each file.



Reading a File to the Display

Refer to the COPY command for other ways to read data.

Example: Display the data that you recorded in the previous example.

Step 1. Assign the filename used in the previous example as the source. Press COMMAND , ASSIGN , more , S:file , EXERCISE, RETURN .

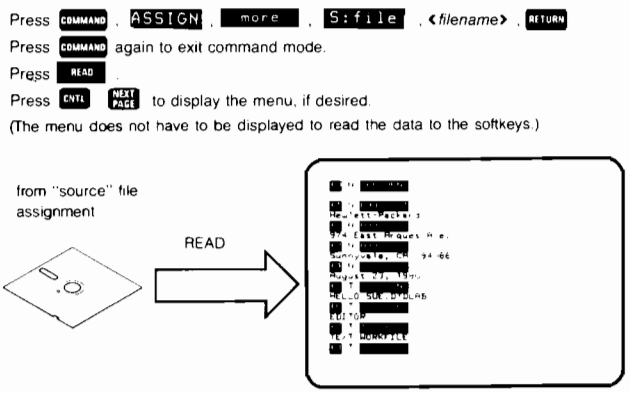
Step 2. Press READ . The data in the file will be displayed on the screen.

Reading Data to the f1 — f8 Softkey Menu

The f1 through f8 function key assignments that have been recorded previously on a disc file may be read to the Softkey Menu. This is an easy way to reload the softkeys at power turn-on, after a hard reset, or if the softkeys are already loaded with another set of functions.

You do not have to display the menu to load the softkeys. Assign the disc file containing the soft key assignments as the "Source", then press READ .

NOTE: You may also use the *Copy command* to load the softkeys.

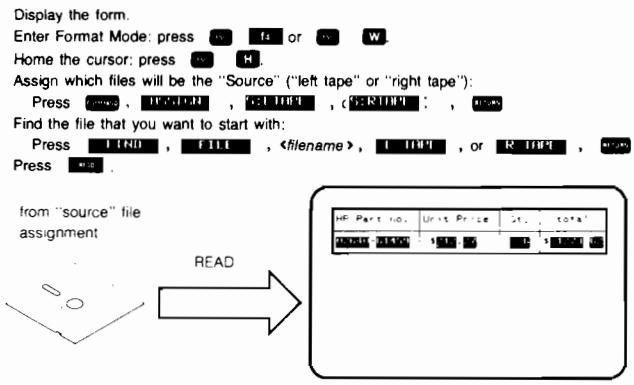


Reading Stored f1 through f8 **Softkey Assignments** to the Softkey Menu

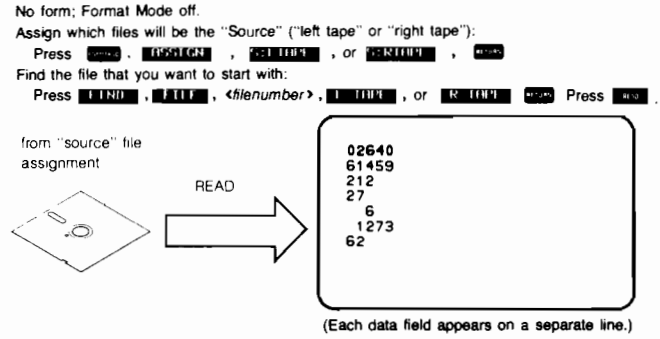
Reading to the Display in Format Mode

Data that has been previously recorded on a diskette in Format mode may be read back to the display. Generally, it is a good idea to display the form from which the data was read, then read the data into it. Be sure to enter Format Mode CNTL f4 or ESC , W before reading the data. Again, you must position the cursor to the first field of the form (ESC , H) for the first read operation. Subsequent reads will leave the cursor in the correct position.

You may read the data to the display without the form present; the data for each field will be displayed on a separate file.



Reading Stored Forms Mode Data to the Form on the Display



Reading Stored Forms Mode Data to the Display (no form)

APPENDING DATA

You can append data to a file providing that enough space is left in the file for the data. If the file is the *last* file on the volume, there should be no problem; no other files are blocking the way for recording the data. However, a file that is *not* the last file on the volume may not have enough space left to accommodate the data to be appended, unless the file was "created" to be a specific length to leave enough room for the appending data. (Refer to "Creating Files of Specific Length".) If the file was not created to be a specific length, the most data that could be added is three lines, depending upon how much space is left in the last sector of the file. If not enough room exists, the **End of disc file** message is displayed when the sector becomes full.

Example: Append data to the "EXERCISE" file used previously in the paragraph titled "Recording Data From the Display".

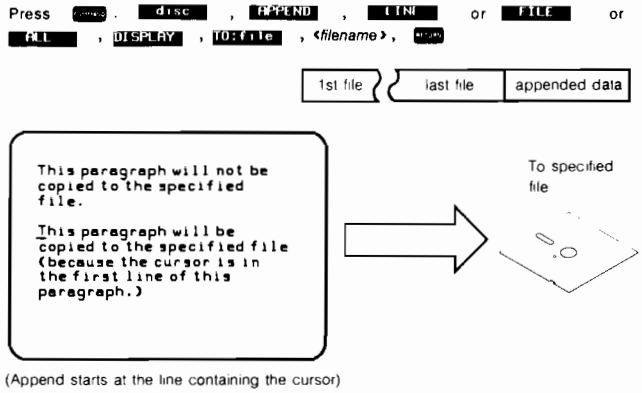
Step 1. Type some data on the display.

Step 2. Position the cursor to the beginning of the data.

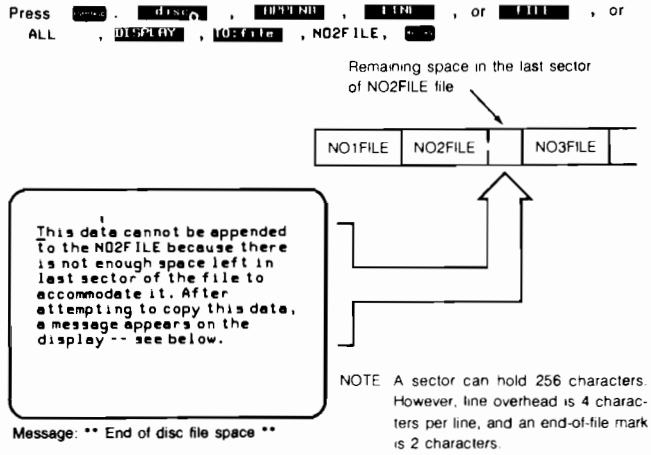
Step 3. Press [COMMAND], **disc**, **APPEND**, **FILE**, **DISPLAY**, **TO:file**, EXERCISE, [RETURN].

Step 4. Verify that the data has been appended. Assign "EXAMPLE" as the source file. Press **ASSIGN**, **more**, **S:file**, EXAMPLE, [RETURN].

Step 5. Press [READ].



Appending Data to the Last File on a Diskette



Attempting to append data to a file other than the last file on a Diskette.

COPYING ASCII DATA

Data may be copied between disc files, printers, and the display by a method other than the [READ] and [RECORD] keys. The *Copy Command* allows you to specify the "from" file or device and "to" file or device for the data and initiate the copy operation in one command sequence. The *assigned* "Source" and "Destination" discussed previously are not altered.

Refer to "Data Operations" in Section 5 for detailed information on use of the Copy and Transfer commands.

Between Disc File and Display or Printer

When copying from the display, remember that the copy starts at the beginning of the line containing the cursor. If "LINE" is specified as the amount of data to be copied. the line containing the cursor is copied. If "FILE" is specified, the copy operation stops at the bottom line of

the display. If "ALL" is specified, the copy operation stops at the end of the current display workspace.

When copying from a disc file either "FILE" or "ALL" will copy the entire contents of the file. If "LINE" is specified, only the *first line* of the file will be copied. (The "LINE" specification is intended to be used with tape emulation.)

Example: Type some data on the display and store it in a file named "EXERCISE1".

Step 1. Type some data on the display.

Step 2. Press [↖] to home the cursor.

Step 3. Press [COMMAND], COPY, ALL, DISPLAY, TO:FILE, EXERCISE1, [RETURN].

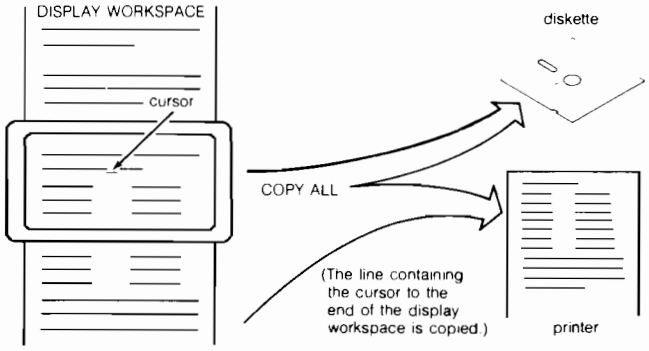
Step 4. Verify the copy operation by showing the disc files on the display. Press SHOW, FILES, [RETURN].

```
Volume - SECT8

  File        File    Start    Size        Create          Create
  Name        Type    Address  (sectors)   Date (M/D/Y)    Time
  ---------   ------  -------  ---------   ------------    --------
  EXERCISE    Ascii     32        1         9/14/80          3:55 PM
  EXERCISE1   Ascii     33        1         9/15/80         11:09 AM
```

Step 5. Press [COMMAND] again to exit command mode.

Press [COMMAND], COPY, ALL, DISPLAY, TO:File, <*filename*> and/or EX PRNTR or ALL PRNTR <*address*> or TO: FILE <*address*>, [RETURN]



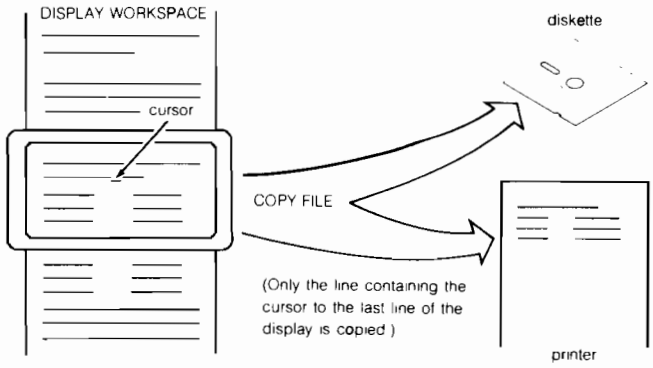
Copying "All" from Display to a Discfile and/or Printer

Example: Copy the data stored in the previous example to the display.

Step 1. Press [COMMAND], COPY, FILE, EXERCISE1, to, DISPLAY, [RETURN].

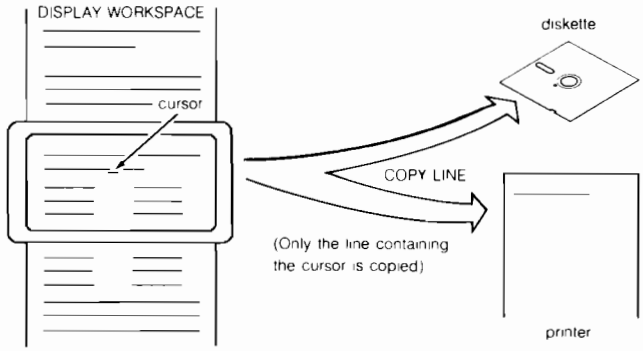
Step 2. Press [COMMAND] again to exit command mode.

Press [COMMAND], COPY, FILE, DISPLAY, TO:File, <*filename*> and/or EX PRNTR or ALL PRNTR <*address*> or TO: FILE <*address*>, [RETURN]



Copying a File from Display to a Discfile and/or Printer

Press [COMMAND], COPY, LINE, DISPLAY, TO:File, <*filename*> and/or EX PRNTR or ALL PRNTR <*address*> or TO: FILE <*address*>, [RETURN]



Copying a Line from Display to a Discfile and/or Printer

From Disc File to Disc File

When copying a disc file to another disc file on the same volume, the "to" filename cannot be the same as the "from" filename. (Remember that two files with the same name cannot exist on the same volume.)

Example: Copy disc file "EXERCISE" to disc file "EXERCISE2".

Step 1. Press [COMMAND], COPY, FILE, EXERCISE, to, TO:file, EXERCISE2, [RETURN].

Step 2. Verify that the file has been copied. Press SHOW, FILES, [RETURN].

```
Volume - SECT8

  File        File    Start    Size        Create          Create
  Name        Type    Address  (sectors)   Date (M/D/Y)    Time
  ---------   ------  -------  ---------   ------------    --------
  EXERCISE    Ascii     32        1         9/14/80          1:43 PM
  EXERCISE1   Ascii     33        1         9/15/80         10:15 AM
  EXERCISE2   Ascii     34        1         9/15/80          2:12 PM
```

Step 3. Press COMMAND again to exit command mode.

If you have two disc drives, then copying a disc file with the same name on the other volume is allowed. However, you must include the volume name of the diskette mounted on disc drive #2. The volume name is included by typing

<filename>;<volumename>,

or

<filename>:<disc drive number>.

Note that the semicolon (;) or colon (:) separates the filename from the name or number (with no spaces).

Example: Copy disc file "EXERCISE" from volume name "SECT8" on disc drive #1 to volume name "NAME2" on disc drive #2. (Note that the volume name must be included when specifying a file on disc drive #2.)

Step 1. Press COMMAND , COPY , FILE , EXERCISE, to , TO:file , EXERCISE;NAME2, RETURN .

Step 2. Verify that the file has been copied. Press SHOW , FILES , RETURN .

Step 3. Press COMMAND again to exit command mode.

Duplicating Volumes

If you have two disc drives, duplicates of entire volumes can be made. The "to" diskette must be formatted, but it must not be named. A *Show Volumes command* will give the status of the diskette. The possible results of the *Show Volumes command* is given below:

```
                VOLUME NAME          HP-IB ADDRESS
                -----------------    -------------

                Local   Disc         Start of    Space  Sector  Create       Create   Write
                Volume  Type#Unit    Free Space  Left   Stagger Date (M/D/Y) Time     Protect
"nonfmt" is     ------- ---------    ----------  -----  ------- ------------ -------- -------
an unformatted  nonfmt  mfD#2              0     1056      9      6/21/81    8:38 AM    NO
diskette

                VOLUME NAME          HP-IB ADDRESS
                -----------------    -------------

"nolabel" is    Local   Disc         Start of    Space  Sector  Create       Create   Write
a formatted     Volume  Type#Unit    Free Space  Left   Stagger Date (M/D/Y) Time     Protect
unnamed         ------- ---------    ----------  -----  ------- ------------ -------- -------
diskette        nolabel mfD#1              0     1056    w 9       8/ 5/81    11:04 AM   NO

                VOLUME NAME          HP-IB ADDRESS
                -----------------    -------------

                Local   Disc         Start of    Space  Sector  Create       Create   Write
                Volume  Type#Unit    Free Space  Left   Stagger Date (M/D/Y) Time     Protect
                ------- ---------    ----------  -----  ------- ------------ -------- -------
                SALES   mfD#1              496     650      9       8/23/81    3:12 PM    NO

                This is a named ("SALES"), formatted diskette
```

If the diskette is "nonsif", you are ready to copy a volume to it.

If it is "nonfmt", it needs to be formatted; execute a *Format Volume command* (the STAGGER parameter may be omitted).

Example: Format the diskette in disc drive #1.

Step 1. Press COMMAND , disc , next , FORMAT , 1, RETURN .

The message Disc FORMAT in progress will be displayed for approximately 45 seconds, then the message Disc VERIFY in progress will be displayed for approximately 45 seconds.

Step 2. Check the formatting by pressing SHOW , VOLUMES , RETURN . Compare the results with the above explanation.

Step 3. Press COMMAND again to exit command mode.

If the diskette is named, you may PURGE the volume, leaving it formatted and unnamed. However, before purging, you should be sure that the data on the disc is not valuable.

Example: Make a duplicate copy of the volume "SALES" on disc drive #2 to disc drive #1.

Step 1. Insert an unnamed, formatted diskette into disc drive #1.

Step 2. Press COMMAND , COPY , VOLUME , SALES, to , 1, RETURN .

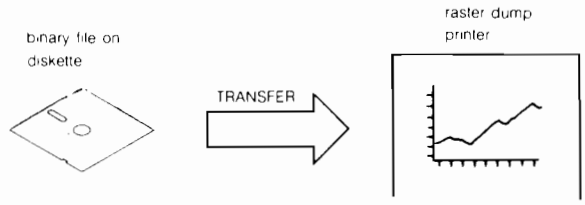
Step 3. Verify the contents of the volume by pressing SHOW , VOLUMES , RETURN .

Step 4. Press COMMAND again to exit command mode.

COPYING BINARY DATA

Either the *Transfer* or *Copy command* can be used to copy data, in *eight-bit binary form* between the display, disc files, and printers. Typically, eight-bit binary data is graphics image data from graphics-type terminals, or programming code in binary form.

Press , next , next , TRANSFER , FILE , *<filename>*, to , or , *<address>*, or , *<address>*, RETURN

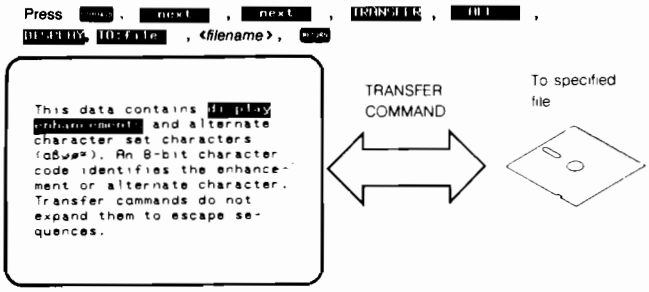


Transferring graphics image data from a discfile to a raster dump printer

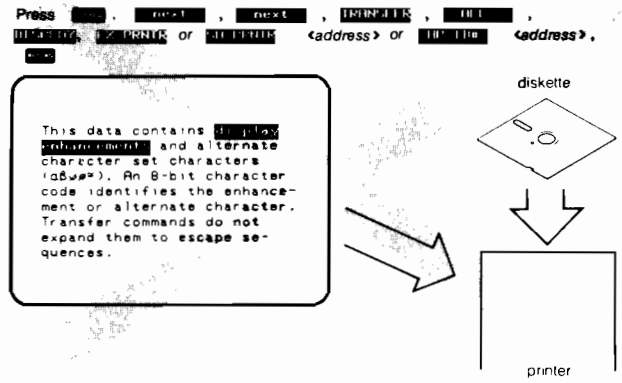
Printers to which graphics data can be transferred are limited to printers with a raster-dump capability. Refer to "Data Operations" in Section 5 for detailed information on use of the *Transfer* and *Copy* commands.

From Display to a Disc File and/or Printer

Display enhancement data (i.e., half-bright, underline, blinking), shift-in and shift-out characters for alternate character sets, etc., are stored as an eight bit character in display memory. The seven other bits are for a normal ASCII character. When the *Copy command* is used to copy the information to a disc file, these data elements are expanded to the corresponding escape sequences. For example, the half-bright character would expand to ESC&dH. However, if the *Transfer command* were used, the half-bright character would be stored on the disc file, instead of the expanded escape sequence. There is no problem when "transferring" data between the display and a diskette. The problem arises when "transferring" the binary data to a printer; in this case, the *Copy command* must be used.



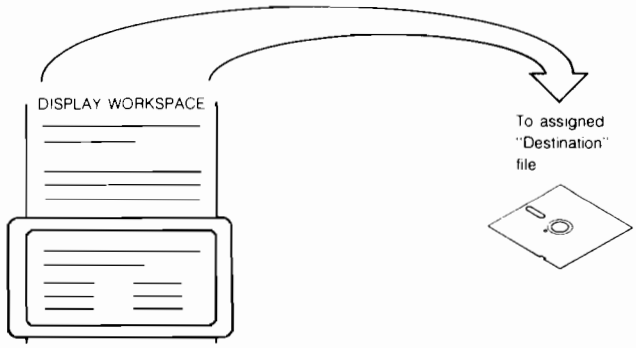
Using the Transfer Command to Copy Display Enhancements and Alternate Character Sets to a Discfile



Do not use the Transfer Command to copy display enhancements and alternate character sets from a discfile or display to a printer.

USING EDIT MODE TO EDIT YOUR FILES

Your disc files may be edited (corrected or updated) using Edit Mode. Edit Mode is a feature of the terminal that causes the top of the present display workspace to roll off to a specified file when the workspace becomes full. (i.e., When the workspace is full and a new line of data is entered, the top line of the workspace is automatically sent to the specified "to" file instead of being lost.) This allows you to enter data onto the screen and have it roll off to the "to" file, or to read data from a specified "from" file to the screen, edit the data, and then record it on the "to" file.



When the current display workspace becomes full in Edit Mode, adding more data will cause the top of the workspace to roll-off to the "destination" file.

Edit Mode Operation

While in Edit Mode, you may enter Text Preparation Mode or Forms Design Mode to allow you to generate or modify your data more easily. (Text Preparation Mode and Forms Design Modes are discussed in the Word/47 manual.

While in Edit Mode, the editing technique is:

(1) read data from a source file to the display (using the READ key),

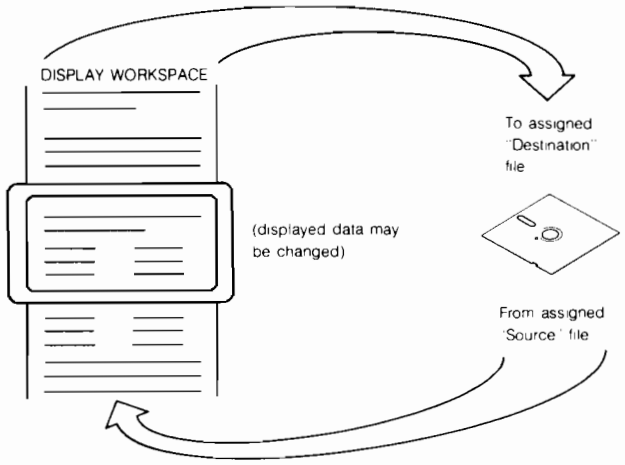
(2) stop the read operation at any time (by pressing the RETURN key),

(3) make corrections to the data on the display,

(4) continue the read-stop-correction process until all the corrections have been made.

(5) Press RECORD to transfer any remaining data in the source file and on the display to the destination file.

Pressing [redacted] copies the data from the previously assigned "Source" file to the display workspace. Pressing [redacted] stops the read operation to allow corrections on the display. The read operation may continue by pressing [redacted] again. When the workspace becomes full, data at the top of the workspace rolls off to the previously assigned "Destination" file. When you have finished correcting the data in the "source" file, press [redacted] to put the remaining data on the display (and the "source" file) into the "destination" file. When specifying filenames on a diskette mounted in a second drive, be sure to include the volumename separated by a semicolon (i.e., <filename>; <volumename>).



Editing Discfile Data Using Edit Mode

Edit Mode may be entered by any one of three methods, depending upon which method would suit your needs:

(1) Enable Edit Mode command sequence (using the Source and Destination assignments).

(2) Edit File command sequence (specifying files without altering the Source and Destination assignments).

(3) Update File command sequence.

NOTE: The Append command sequence may be used to add data to the end of a disc file.

Edit Mode Using the "Source" and "Destination" Assignments

This method is useful for editing Tape Emulator files (i.e., where the source and destination can remain constant for multiple file edits). For example, suppose you have a group of Tape Emulator files for the "left tape" which need editing. You have no Tape Emulator Files for the "right tape" on the disc. Therefore, you assign the left tape as the "Source" and the right tape as the "Destination".

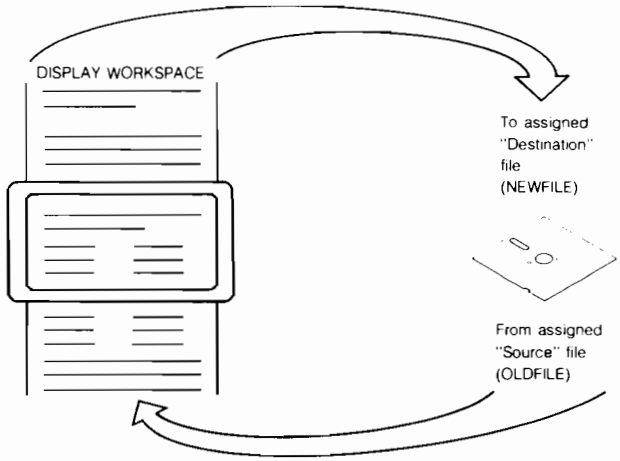
This method uses the assigned "Source" and "Destination" files. The *Show Assignments command* will display the current assignments. The power-on and reset defaults are:

Source: Left Tape
Destination: Right Tape

To enter the Edit Mode, press:
COMMAND , next , next , edit , ENABLE , RETURN .

The EDIT indicator above the f8 key will turn on.

Press [redacted] , [redacted] , [redacted] , [redacted] , <*OLDFILE*>, [redacted] .
Press [redacted] , [redacted] , [redacted] , <*NEWFILE*>, [redacted] .
Press [redacted] , [redacted] , [redacted] , [redacted] , [redacted] .




```
USER NAME      CURRENT ASSIGNMENTS
-----------    -------------------
SOURCE         OLDFILE
DESTINATION    NEWFILE
```

Edit Mode Using "Source" and "Destination" Assignments

Edit File Command

The *Edit File command* allows you to specify the "source" and "destination" files while entering Edit Mode. This does not alter the "Source" and "Destination" in the Assign table.

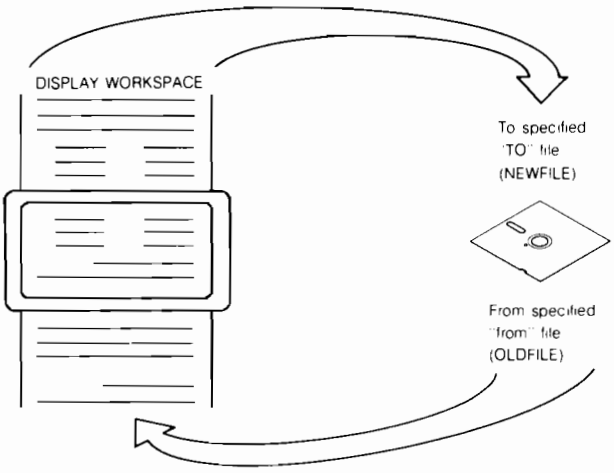
The Edit File command sequence is entered by pressing:
COMMAND , next , next , edit , FILE ,
<*old disc file name*>, to , TO:file ,
<*new disc file name*>, RETURN .

The EDIT indicator above the f8 key will turn on.

After pressing FILE in the command sequence, five "source" selections are available: left tape, right tape, display, an HP-IB device address, or enter a disc file name. You would enter the old disc file name here (i.e., the name of the file that you want to edit).

After pressing to in the command sequence, eight "destination" selections are available: left tape, right tape, display, TO:file, external printer, shared printer, an HP-IB address, and graphics memory. You would specify the new disc file name after pressing TO:file . You may select additional destinations by pressing the other keys.

Press COMMAND , next , next , edit , Fill , ‹OLDFILE›, to , TO:file , ‹NEWFILE›, RETURN .



The current 'Source' and 'Destination' assignments are unchanged

```
USER NAME      CURRENT ASSIGNMENTS
-----------    -------------------
SOURCE         LEFT TAPE
DESTINATION    RIGHT TAPE
```

Edit Mode Using Edit File Command

PURGING AND UNPURGING FILES AND VOLUMES

Purging a file or volume means to remove it from the active file directory, thereby making it inaccessible. When a *Show Files command* is executed, only the active files (i.e., only those that are not purged) are listed. A *Show Purge command* shows the purged files on the volume.

Purging allows you to keep a "clean" volume of usable data. Should you make a mistake, and purge the wrong file or volume, you can recover by the *Unpurge command*. However, if you have executed a *Pack Volume command* after purging the file(s), it is not possible to recover the data.

Purging a File

Suppose that you have a file named "EXAMPLE3" in your volume directory, and would like to purge that file.

Show Files listing before purge:

```
Volume = MYFILE

  File        File     Start     Size        Create          Create
  Name        Type     Address   (sectors)   Date (M/D/Y)    Time
  ---------   ------   -------   ---------   ------------    --------
  EXAMPLE1    Ascii       32        12         3/ 4/80       12:31 PM
  EXAMPLE2    Ascii       44        14         3/ 4/80        9:28 AM
  EXAMPLE3    Ascii       58        29         3/ 4/80       11:59 AM
  EXAMPLE4    Ascii       87        39         3/ 4/80        4:02 PM
  EXAMPLE5    Ascii      126        39         5/15/80        1:38 PM
```

Press COMMAND , disc , PURGE , FILE , EXAMPLE3, RETURN .

Show Files listing after purge:

```
Volume = MYFILE

  File        File     Start     Size        Create          Create
  Name        Type     Address   (sectors)   Date (M/D/Y)    Time
  ---------   ------   -------   ---------   ------------    --------
  EXAMPLE1    Ascii       32        12         3/ 4/80       12:31 PM
  EXAMPLE2    Ascii       44        14         3/ 4/80        9:28 AM
  EXAMPLE4    Ascii       87        39         3/ 4/80        4:02 PM
  EXAMPLE5    Ascii      126        39         5/15/80        1:38 PM
```

A Show Purged command sequence will show the purged files.

Press SHOW , PURGE , RETURN . Press COMMAND again to exit command mode.

```
Volume = MYFILE

  File        File     Start     Size        Create          Create
  Name        Type     Address   (sectors)   Date (M/D/Y)    Time
  ---------   ------   -------   ---------   ------------    --------
  EXAMPLE3    Purged      58        29         3/ 4/80       11:59 AM
```

Unpurging a File

Now, let's restore the file to active status.

Press COMMAND , disc , UNPURGE , FILE , EXAMPLE3, RETURN .

If you unpurge a file that has the same name as an active file, the message File name already exists will be displayed. You must rename the active file (see "Renaming Files"), then unpurge the purged file.

Purging a Volume

When a volume contains no useful files, you can purge the volume, then execute a *Create Volume command* to give the volume a new name. *Should you wish to unpurge the volume, it must be done before the Create Volume command is executed.* Once the *Create Volume command* is executed, the data may not be recovered.

Example: Purge the volume named "MYFILE".

Press COMMAND , disc , PURGE , VOLUME , MYFILE, RETURN .

A *Show Volumes command* shows "nolabel" for the volume name.

```
VOLUME NAME          HP-IB ADDRESS
------------------   -------------

Local    Disc        Start of    Space  Sector   Create          Create     Write
Volume   Type#Unit   Free Space  Left   Stagger  Date (M/D/Y)    Time       Protect
-------  ----------  ----------  -----  -------  ------------    ---------  -------
nolabel  mfD#1             0     1056      9       6/12/81      10:46 AM     NO
```

Unpurging a Volume

Now, let's unpurge the volume.

Press [COMMAND] , disc , UNPURGE , VOLUME , 1, [RETURN] .

A *Show Volumes command* verifies the unpurge (the old volume name is restored).

```
VOLUME NAME          HP-IB ADDRESS
-------------------  -------------

Local   Disc      Start of    Space  Sector   Create        Create    Write
Volume  Type#Unit Free Space  Left   Stagger  Date (M/D/Y)  Time      Protect
------  --------- ----------  -----  -------  ------------  --------  -------
MYFILE  mfD#1          306     750      9      6/12/81     10:46 AM    NO
```

PACKING DATA ON VOLUMES

Packing data on a volume means eliminating the space occupied by the purged files. The active files are moved such that the space occupied by the purged files is overwritten. After packing, the active files are contiguous from directory to the end of the last active file (no purged files between them).

The following shows the active and purged files on volume "MYSTUF".

```
Volume = MYSTUF

  File       File    Start    Size       Create         Create
  Name       Type    Address  (sectors)  Date (M/D/Y)   Time
  ---------  ------  -------  ---------  ------------   --------
  FILENAME2  Ascii       44        14       6/ 1/80     12:01 PM
  FILENAME5  Ascii       71        17       7/14/80      3:45 PM

Volume = MYSTUF

  File       File    Start    Size       Create         Create
  Name       Type    Address  (sectors)  Date (M/D/Y)   Time
  ---------  ------  -------  ---------  ------------   --------
  FILENAME1  Purged      32        12       5/15/80      9:37 AM
  FILENAME3  Purged      66         5       6/ 7/80      2:40 PM
  FILENAME4  Purged      66         5       6/ 7/80      2:40 PM
```

To pack the volume, press [COMMAND] , disc , next , PACK , MYSTUF, [RETURN] .

The message Disc PACK in progress from (n) to (m) is displayed. (<n> is the old sector number and <m> is the new sector number.)

Files before PACK:

FILENAME1	FILENAME2	FILENAME3	FILENAME4	FILENAME5
(purged)		(purged)	(purged)	

Files after PACK:

FILENAME2	FILENAME5			

Note the Start Addresses of active files after the *Pack command* is executed.

```
Volume = MYSTUF

  File       File    Start    Size       Create         Create
  Name       Type    Address  (sectors)  Date (M/D/Y)   Time
  ---------  ------  -------  ---------  ------------   --------
  FILENAME2  Ascii       32        14       6/ 1/80     12:01 PM
  FILENAME5  Ascii       46        17       7/14/80      3:45 PM
```

RENAMING FILES

File names may be changed by the *Rename File command*. This command is useful when you want to generate or activate a file with a name that already exists on the volume. (It allows you to change the name of the existing file.) If you attempt to generate or activate a duplicate filename, the message File name already exists will appear in the message line of the display.

For example, an active filename "YOURNAME" already exists on the volume (a *Show File command* reveals this), and you do not want to purge this file. You want to copy some data on the display to a new file "YOURNAME", but two files with the same name cannot be active on the same volume. Before the copy can be made, the current "YOURNAME" file must be renamed (using the *Rename File command*).

Press [COMMAND] , disc , RENAME , FILE , YOURNAME, to , NEWNAME, [RETURN] .

Renaming is also necessary when unpurging a file with the same name as an active file. For example, the *Show Files command* lists the following:

```
Volume = MYFILE

  File       File    Start    Size       Create         Create
  Name       Type    Address  (sectors)  Date (M/D/Y)   Time
  ---------  ------  -------  ---------  ------------   --------
  EXAMPLE1   Ascii       32        12       3/ 4/80     10:12 AM
  EXAMPLE2   Ascii       44        14       3/ 4/80     10:48 AM
  EXAMPLE3   Ascii       58        29       3/ 4/80     11:21 AM
  EXAMPLE4   Ascii       87        39       3/ 4/80     11:55 AM
  EXAMPLE5   Ascii      126        39       5/15/80      2:34 PM
```

The *Show Purged command* lists the following:

```
Volume = MYFILE

  File       File    Start    Size       Create         Create
  Name       Type    Address  (sectors)  Date (M/D/Y)   Time
  ---------  ------  -------  ---------  ------------   --------
  EXAMPLE1   Purged     165        12       5/15/80     10:12 AM
  EXAMPLE2   Purged     177         8       6/ 6/80     10:48 AM
  EXAMPLE3   Purged     185         1       6/ 7/80     11:21 AM
```

Note that filenames EXAMPLE1, EXAMPLE2, and EXAMPLE3 exist as unpurged and purged files. Suppose that you wanted to unpurge EXAMPLE1. You must change the name of the active file because two active files with the same name cannot exist. So you decide to rename the active file to "EXAMPLE1A".

Step 1. Press [COMMAND] , disc , RENAME , FILE , EXAMPLE1, TO , EXAMPLE1A, [RETURN] .

Step 2. Now, unpurge file EXAMPLE1. Press disc , UNPURGE , FILE , EXAMPLE1, [RETURN] .

Step 3. Verify the list of current files. Press SHOW , FILES , [RETURN] .

Step 4. Press [COMMAND] again to exit command mode.

```
Volume - MYFILE

File        File    Start    Size       Create          Create
Name        Type    Address  (sectors)  Date (M/D/Y)    Time
---------   -----   -------  ---------  ------------    --------
EXAMPLE1A   Ascii      32       12        3/ 4/80       10:12 AM
EXAMPLE2    Ascii      44       14        3/ 4/80       10:48 AM
EXAMPLE3    Ascii      58       29        3/ 4/80       11:21 AM
EXAMPLE4    Ascii      87       39        3/ 4/80       11:55 AM
EXAMPLE5    Ascii     126       39        5/15/80        2:34 PM
EXAMPLE1    Ascii     165       12        5/15/80       10:12 AM
```

COMPARING FILES AND VOLUMES

The *Compare command* makes a byte-by-byte comparison of the data contained in the specified files, display, or volumes. If a match is not made in any record, execution of the command stops. One of the following messages will appear in the message line of the display:

Compare File Messages:

`DIFFERENT LENGTH RECORDS, RECORD#<n>` (<n> is an integer.)

`DIFFERENCE IN BYTE#<n>, RECORD#<n>`

`DIFFERENCE IN RECORD TYPE, RECORD#<n>`

Compare Volume Message:

`Data on volumes did not compare <n>` (<n> is the sector no.)

After a mismatch is found, the comparison may be continued for Tape Emulated files by re-executing the command. (A file may be converted to a Tape Emulator file by the Rename command.) If a mismatch is found on disc files that are not tape-emulated, the comparison cannot be continued; you must rename the file(s) to a tape-emulated file(s), then start the compare again.

Comparing Files

In the following example, two disc files are compared. If a difference occurs, each file is renamed to a Tape Emulator file (one left tape and one right tape). The compare is initiated again. When the message is displayed, each file is positioned back one record (where the difference occurred), and the two different records displayed on the screen.

Example: Compare a disc file named "ORDERDATA" with data on disc file named "ORDERS".

Step 1. Press COMMAND, next, next, COMPARE, FILE, ORDERDATA, file:TO, ORDERS, RETURN.

Step 2. If a message appears, the difference may be found by renaming each file to a Tape Emulator file, then executing the Compare File command again, as given below.

Step 3. Rename "ORDERDATA" to "CTUL001" and rename "ORDERS" to "CTUR001".

Press DELETE LINE to clear the command line, then press next, top, disc, RENAME, FILE, ORDERDATA, TO, CTUL001, RETURN.

Press disc, RENAME, FILE, ORDERS, TO, CTUR001, RETURN.

Step 4. Re-execute the compare. Press next, next, COMPARE, FILE, L TAPE, R TAPE, RETURN.

Step 5. When the compare file message appears, press DELETE LINE to clear the command line.

Step 6. Skip −1 line on each file. Press next, next, SKIP, -1, LINES, L TAPE, R TAPE, RETURN.

Step 7. Read each line to the display to make a visual comparison.

Press COPY, LINE, L TAPE, DISPLAY, RETURN.

Press COPY, LINE, R TAPE, DISPLAY, RETURN.

Press COMMAND again to exit command mode.

Comparing Volumes

Entire volumes may be compared by using the *Compare Volume command.* (This requires two disc drives, one for each volume.) This command may be used after the *Copy Volume command* to verify that the data was copied exactly from one volume to the other.

Example: Compare volumes named "MONTH".

Step 1. After the diskettes are mounted, execute the Compare Volumes command.

Press COMMAND, next, next, COMPARE, VOLUME, MONTH1, TO, 1, RETURN.

Step 2. If the following message is displayed, `Data on volume did not compare (n)` re-execute the Copy Volume command, then compare the volumes again.

VERIFYING A VOLUME

The *Verify Volume command* checks the CRCs (error checking codes) and the recording tolerances on the diskette. This command should be used if you are having trouble reading files on a particular diskette. The message **Disc VERIFY in progress** is displayed while the verify operation is in progress. If the data does not verify at a sector, the **Disc data did not verify** message is displayed.

A *Verify Volume command* is executed automatically after formatting a diskette.

Example: Verify the data on the diskette mounted in disc drive#1.

Step 1. Press [COMMAND] , **next** , **next** , **verify** , **VOLUME** , 1, [RETURN] .

Step 2. The message **Disc VERIFY in progress** is displayed while the verify operation is in process.

The message **Disc data did not verify** is displayed if any sector does not verify.

THE WELCOME FILE

A WELCOME file is an *Execute File* created by you. One of the features of the terminal is that when the terminal is turned on, it looks automatically for a file named "WELCOME" on the Disc Drive. If the WELCOME file is present, then the terminal *executes* it. (*Execute files* are explained in the Commands section.) A WELCOME file could do one or more of the following:

- load the user-defined soft keys ([f1] through [f8]) with any functions defined by you,
- display a form from a disc file,
- log you onto your account in the computer system,
- display instructions from a disc file,
- configure the terminal through escape sequences.

Example: Create a `WELCOME` file to log you onto an account "`ME.LAB`" in the HP 3000 Computer System, enter the EDITOR subsystem, and get the file named "`MYFILE`".

Step 1. Clear the display ([↖] , [CLEAR DISPLAY]). Enter the following lines onto the display:

NOTE: The ↰ character is displayed by pressing [DISPLAY FUNCTIONS] then pressing [ESC] . Press [DISPLAY FUNCTIONS] again to turn off Display Functions Mode.

```
(blank line) ← to get the colon prompt
HELLO ME.LAB
EDITOR
TEXT MYFILE
```

Step 2. Record or copy the contents of the display to a disc file named "`LOGON`". Press [↖] , [COMMAND] , **COPY** , **FILE** , **DISPLAY** , **TO:file** , LOGON, [RETURN] .

Step 3. Press [COMMAND] again to exit command mode.

Step 4. Clear the display ([↖] , [CLEAR DISPLAY]), and enter the following onto the display (not into the command line):

```
TELL DISPLAY "↰&k1R"
COPY FILE LOGON TO DATACOMM
```

Step 5. Copy the contents of the display to a disc file named "WELCOME". Press [↖] , [COMMAND] , **COPY** , **FILE** , **DISPLAY** , **TO:file** , WELCOME, [RETURN] .

Step 6. Press **SHOW** , **FILES** , [RETURN] to verify the presence of the files on the volume.

Step 7. Turn off power, then turn on power. The terminal's WELCOME file should log-on to the HP 3000 Computer System, get the Editor subsystem, and text in `MYFILE`.

CARTRIDGE TAPE EMULATION

Cartridge Tape Emulation is a feature which enables the terminal to operate on disc files in response to tape commands, treating the disc files as though they were files on tape. The tape commands may originate from the keyboard or from escape sequences sent from a host computer. This feature is useful in data entry applications when Forms Mode data is stored on a diskette in local mode. Also, it provides backward compatibility with other 264X terminals that use cartridge tape units (i.e., previously written programs to control the tape units can also be used to drive the Cartridge Tape Emulator). Program control of the cartridge tape emulator through escape sequences is discussed in the Commands section.

The paragraphs that follow discuss the conditions under which cartridge tape emulation becomes active and how to use tape commands with cartridge tape emulated files from the keyboard. Table 6-1 lists which tape operations are supported by the emulator.

Table 6-1. Cartridge Tape Commands Supported by the Emulator

Fully-Supported % Commands	Partially-Supported % Commands	Not Supported % Commands
Compare	Find End-of-Data	Condition
Copy	Mark File	Test
Find File	Show Tapes	
Rewind	Skip Lines	
Transfer		
Verify		

Disc files are assigned in groups for susceptibility to tape commands. A group can consist of 1 file, 1 through 9 files, 1 through 99 files, or 1 through 255 files. To make a group of files susceptible to CTU Emulator commands, two requirements must be met:

1. The disc files must be named according to CTU Emulator requirements.
2. The group of files must be assigned as the left CTU (CTUL) or the right CTU (CTUR) using the Assign command.

CTU Emulator Naming Convention

If a group consists of more than one file, the name for any file in the group must consist of the group name, common to all files in the group, and a file number. The file numbers start with 1 and run consecutively. For example, if a group is to be composed of up to nine files with the group name of MYGRUP, the files must be named MYGRUP1, MYGRUP2, MYGRUP3... up to the last file in the group. If MYGRUP is to consist of up to 99 files, the files must be named MYGRUP01, MYGRUP02, MYGRUP03, ... MYGRUP99.

If the group consists of only one file, it doesn't need a file number; the name MYGRUP suffices.

Assigning the Designator "CTUL" or "CTUR" to a Group of Files

When a CTU designator (CTUL or CTUR) is assigned to a group of files, questions marks (?) are used, embedded in or following the group name in the Assign command, to indicate the size of the group. One question mark indicates a group of two to nine files, two question marks indicates a group of two to 99 files, and three question marks indicate a group of up to 255 files. Embedding the question marks in the group name is allowed.

Example: Assign the designator CTUL to the single file MYGRUP:

Press COMMAND , ASSIGN , more , more , CTUL , MYGRUP, RETURN .

Example: Assign CTUR to the group of files MYGRUP1 through MYGRUP9:

Press COMMAND , ASSIGN , more , more , CTUR , MYGRUP?, RETURN .

Example: Assign CTUL to the group of files MYGRUP001 through MYGRUP255:

Press COMMAND , ASSIGN , more , more , CTUL , MYGRUP???, RETURN .

Example: Assign CTUR to the group of files MY01GRUP through MY99GRUP:

Press COMMAND , ASSIGN , more , more , CTUR , MY??GRUP, RETURN .

Purging and Unpurging Tape Emulator Files

AUTOMATIC PURGING. The file at the current position of the "tape" will be purged automatically whenever a command to write data to a "tape" is executed (such as Copy, Transfer, or record operation). For example, suppose that three CTU emulator files (CTUL-001, CTUL002, and CTUL003) exist on the diskette and that the current position of the "tape" is at the beginning of the second file (CTUL002). Copying data to the "left tape" would purge automatically the current CTUL002, and write a new CTUL002 at the next available sector on the disc.

If you wanted to unpurge the old CTUL002 file to recover the data, you would need to rename the *current* CTUL002 to another name, then execute an *Unpurge File command.*

USING THE PURGE FILE COMMAND. If you purge a CTU emulator file, the end-of-data mark will be updated to the last contiguously numbered file in the sequence. The higher numbered CTU emulator files can only be accessed by a disc command. (That is, specifying the file by name in the copy operation.) Refer to "Find End-of-Data" described below.

Fully Supported Tape Commands

COMPARE COMMAND. The *Compare command* makes a byte-by-byte comparison of the data between two "tapes" or between a "tape" and the display. A line, a file, or all may be compared. For example, if you wanted to compare a file, position the "tapes" to the files to be compared by the *Find File command*, then execute the *Compare File command.* If you wanted to compare a line, position the "tape(s)" by the *Find File* and/or *Skip Lines commands*, then execute the *Compare Line command.*

COPY COMMAND. The *Copy command* copies data from one "tape" to the other tape, display, and/or a printer. A line, a file, or all may be copied. The copy operation starts at the present "position" of the "from" tape and copies to the present position of the "to" tape. Therefore, remember to position the "to" tape before initiating the copy operation; if you don't, you will purge the current file on the "to" tape, and create a new file with the same number. If no CTU files presently exist on the "to" tape, one will be created. If you accidentally purge a file, rename the newly created file; then, execute an *Unpurge File command* specifying the name of the purged file.

FIND FILE. The *File File command* locates a tape emulator file by number. For example, "FIND FILE 6 ON LEFT TAPE" locates the beginning of the file *xxxxxx*L006 on the diskette. (*xxxxxx* is the current "CTU" assignment in the assign table.) Relative file positioning may be executed by typing a sign before the number. For example, assume that the "left tape" is positioned at the beginning of file 6 as in the previous example; "FIND FILE -4 ON LEFT TAPE" locates the beginning of file 2 (6 − 4 = 2).

REWIND. The *Rewind command* closes any open tape emulator file and positions the specified "tape" at the beginning of file 1. A file will be open if the previous tape operation was a write operation, or if a tape emulator file was created by entering Edit mode (the assigned destination).

After a write operation has been performed, be sure to "rewind the tape" before removing the diskette; otherwise, the file will not be closed, and the directory will indicate a zero length file. If a *Copy command* is attempted using the disc file name (e.g., CTUL006), the message "Improper directory data" will be displayed.

TEST TAPES. The *Test Tapes command* writes a worst case data pattern ("Z") on "left tape" emulator file 1 and on "right tape" emulator file 1. Before executing this command, be sure that no valuable data is present as a tape emulator file on the diskette. Data may exist as a regular disc file without danger of being overwritten.

TRANSFER. The *Transfer command* copies binary data to or from tape emulator files when the left tape or the right tape is specified.

VERIFY. The *Verify command* turns on a write-backspace-read mode. If data is copied or transferred to a tape emulator file, a sector is written to the file, backspaced to the beginning of the sector, then read and compared with a copy of initial sector. This feature is desirable when data integrity is important.

Partially Supported Tape Commands

FIND END-OF-DATA. The *Find End-of-Data command* locates the end of the highest contiguously-numbered tape emulator file. If a tape emulator file sequence has been broken by purging a file within the sequence, then the end-of-data will be at the end of the highest contiguously-numbered file. You may add a file to preserve the entire sequence by "finding" the number of the purged file, then copying data to it to create a new one, or you can unpurge the old one. (See below.)

```
Volume - MYFILE

  File       File     Start    Size       Create          Create
  Name       Type     Address  (sectors)  Date (M/D/Y)    Time
  ---------  ------   -------  ---------  ------------    --------
  CTUR001    Ascii      32        5         9/ 3/80        7:03 AM
  CTUR002    Ascii      37        3         9/ 3/80        7:37 AM
  CTUR002    Ascii      40        4         9/ 4/80        3:41 PM
  CTUR005    Ascii      49        6         9/ 5/80       10:58 AM
  CTUR006    Ascii      55        3         9/ 5/80        2:12 PM

Volume - USER

  File       File     Start    Size       Create          Create
  Name       Type     Address  (sectors)  Date (M/D/Y)    Time
  ---------  ------   -------  ---------  ------------    --------
  CTUR004    Purged     44        5         9/ 4/80        4:01 PM
```

The highest contiguously-numbered file is CTUR003; CTUR004 has been purged. Therefore a FIND END OF DATA command would find the end of CTUR003. Note that if file CTUR004 is restored, a FIND END OF DATA command will locate the end of file CTUR006.

Finding end-of-data, then pressing RECORD creates a new file.

MARK FILE. The *Mark File command* writes an end-of-file mark on the diskette to indicate the end of the current file. If Mark File commands are executed repeatedly without writing any data between them, then only the first one is written; the rest are ignored. (When this is tried with cartridge tape, blank files are created.)

SHOW TAPES. The *Show Tapes command* lists the current (or last) tape emulator file which has been accessed. If your diskette contains regular disc files in addition to tape emulator files, and you accessed CTUL004 then accessed one or more regular disc files, the *Show Tapes command* would list the current position of the "left tape" as file number "4". (If no CTU emulator commands have been executed since power on or a hard reset, the *Show Tapes* listing will show file number "0".

If another diskette is inserted into the disc drive, the CTU emulator still remains at the same position. (e.g., If the position of the "tapes" was file 6 on the left tape and file 4 on the right tape before another diskette was inserted into the drive, the position will remain the same when the other diskette is inserted). In other words, the CTU emulator does not automatically "rewind" the left and right tapes when another diskette is inserted into the drive. (The *Show Tapes* listing stays the same, but the pointer actually points to the first files.)

```
DEVICE NAME  FILE NUMBER  INCHES REMAINING
-----------  -----------  ----------------
LEFT TAPE    7            0
RIGHT TAPE   5            0
```

SKIP LINES. The *Skip Lines command* allows you to skip forward or backward a specified number of lines in a tape emulator file. You cannot skip forward or backward to the next or previous file; the "tape" will be positioned either at the end or the beginning of the current file.

If the "tape" is at end-of-data, the *Skip Lines command* is ignored (i.e., it cannot skip forward or backward).

Unsupported Cartridge Tapes Commands

CONDITION TAPES. The *Condition Tapes command* is ignored by the tape emulator.

TEST TAPES. The *Test Tapes command* results in a "NO TAPE IN LEFT DRIVE" message.

PRINTER OPERATIONS FROM THE KEYBOARD

SECTION 7

INTRODUCTION

This section describes how to control printers from the terminal keyboard.

TYPE OF PRINTERS

The table below lists the three types of printers recognized by the terminal's command channel, the interface, and Hewlett-Packard printers associated with each.

PRINTER TYPE	TERMINAL INTERFACE	PRINTER MODEL
External Printer	HP 13250B Serial Printer Interface	HP 2631A/B opt. 051 HP 7310A opt. 050 HP 2601A
	HP 13238A 8-bit Parallel Printer Interface	HP 2631A/B opt. 240 HP 7310A opt. 052 HP 9871A
Shared Printer#	HP 13296A Shared Peripheral Interface	HP 2631A/B opt. 046 HP 2631G std HP 9871A opt. 001 HP 2671A/G HP 2673A HP 9876A
HP-IB#	HP-IB/ROM PCA	HP 2631A opt. 046 HP 2631G std HP 7310A std HP 9871A opt. 001

Note: Shared Printers may also be addressed by HP-IB#.

"External" Printers

"External" printers are those that are connected to the terminal by either the 13238A 8-bit Duplex Register PCA or the 13250B Serial Printer Interface PCA. The EX PRNTR key label is used with this kind of printer.

"Shared" Printers

A "shared" printer is any printer that is connected to the terminal by the 13296A Shared Peripheral Interface *and* that can self-identify. The HP 2631A/B opt. 046, HP 23631G, and HP 9871A opt. 001 printers are examples of this kind of printer. More than one terminal can be connected to the shared peripheral interface network, therefore the term "shared printer". The SH PRNTR key label is used with this kind of printer. (Shared printers may also be addressed by HP-IB#.

Self-identification is verified by executing a *Show Volumes command.* If the printer can self-identify, the show volumes listing will show the printer as "Shared printer n". The printer must be turned on and be on-line. If the printer is connected in a terminal network (i.e., more than one terminal is connected to the printer), then the "system controller" terminal must be turned on also. (How to identify the system controller terminal is described in the Configuration section.

```
VOLUME NAME          HP-IB ADDRESS
-----------------    -------------
Shared printer       7
HP-IB device         6

Local   Disc        Start of    Space  Sector   Create          Create  Write
Volume  Type#Unit   Free Space  Left   Stagger  Date (M/D/Y)    Time    Protect
------  ---------   ----------  -----  -------  ------------    ------  -------
```


Show Volumes Command Listing with a Shared Printer and an HP-IB Printer

Shared printers have header and trailer pages as part of their listings which are defined by the shared printer address, Hello command, terminal date, and terminal time. When data is copied to a shared printer, the first and last pages list these items four times.

```
printer   Set by HELLO      Set by SET DATE      Set by SET TIME
addr      command           command              command
  |     ----------------  ----------------  ----------------
SH#7 USER       .GROUP  ********************  6:47:22 AM
    SH#7 USER       .GROUP  ********************  6:47:22 AM
        SH#7 USER       .GROUP  ********************  6:47:22 AM
            SH#7 USER       .GROUP  ********************  6:47:22 AM
```

Header/Trailer on Shared Printer listings with default User Group and default Date

PRINTER ADDRESS. The Configuration section describes how to set the address on shared printers. The printer address for shared printers should be between 0 and 7 inclusive. If the address is higher than 7 then for device operations the "HP-IB#n" must be specified instead of "SH PRNTR#n". If "HP-IB#n" is specified, the header/trailer pages do not appear on the output listings. You can specify "HP-IB#n" for addresses 0 through 7 if you do not want the header/trailer pages.

HELLO COMMAND. The *Hello command* permits you to specify names for "user.group" on your shared printer listings. Up to 8 characters each may be assigned to "user" and "group". At power turn on or after a hard reset (pressing the [RESET TERMINAL] key twice within 1/2 second) the default is "USER.GROUP".

Example: Define user.group as "PAM.SUPPORT".

Step 1. Press [COMMAND], [next], [HELLO], PAM.SUPPORT, [RETURN].

Step 2. Press [COMMAND] again to exit command mode.

```
SH#7 PAM     .SUPPORT  FRI, MAY  9, 1980  10:43:17 AM
   SH#7 PAM     .SUPPORT  FRI, MAY  9, 1980  10:43:17 AM
      SH#7 PAM     .SUPPORT  FRI, MAY  9, 1980  10:43:17 AM
         SH#7 PAM     .SUPPORT  FRI, MAY  9, 1980  10:43:17 AM
```

Header/Trailer on Shared Printer Listing After Setting Date, Time, and User.Group

BYE COMMAND. The *Bye command* returns the user.group names to the default "user.group".

DATE AND TIME. Date and time on the shared printer listings are set by the *Set Date* and *Set Time* commands.

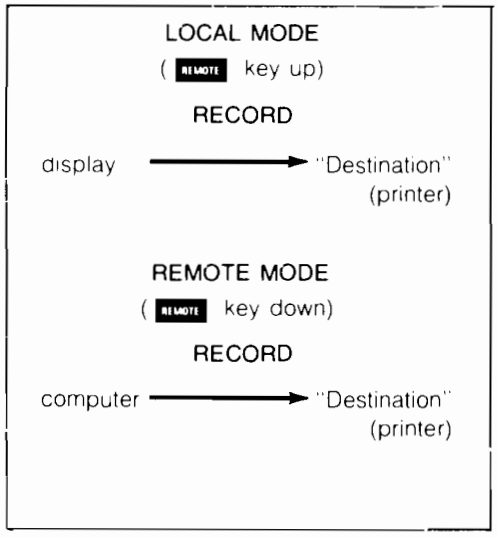
"HP-IB" Printers

An "HP-IB" printer is any printer connected to the terminal by the HP-IB but *cannot* self-identify. The HP 7310A printer is an example of this kind of printer. A *Show Volumes command* will list the printer as "HP-IB device<n>" if the printer is currently a destination device. (The printer must be turned on and be on-line.) The [HP-IB#] key followed by the printer's address are used to specify the printer for device operations.

PRINTER ADDRESS. The Configuration section describes how to set the address on HP-IB printers. The printer address for HP-IB printers can be between 0 through 7.

USING THE RECORD KEY TO PRINT DATA

The [RECORD] key provides a convenient means for printing data from the display or datacomm depending upon whether or not the terminal is in local mode ([REMOTE] key up) or remote mode ([REMOTE] key down). To send the data to a printer using the [RECORD] key, the printer must be assigned as the "Destination" device.



Assigning the Printer as "Destination"

The *Assign command* allows you to specify a printer as the "Destination" device for RECORD operations.

Example: Assign the 9871A printer connected to the terminal via the 13238A interface as the "Destination" device.

Step 1. Press [COMMAND], [ASSIGN], [more], [D.EX PTR], [RETURN].

Step 2. Exit command mode by pressing [COMMAND] again.

Example: Assign the 2631G printer connected to the terminal via the 13296A interface as the "Destination" device. Assume that the printer's address is set to 5.

Step 1. Press [COMMAND], [ASSIGN], [D:SH PTR], 5, [RETURN].

Step 2. Exit command mode by pressing [COMMAND] again.

Example: Assign the 7310A printer connected to the terminal via the 13296A interface as the "Destination" device. Assume that the printer's address is set to 3.

Step 1. Press [COMMAND], [ASSIGN], [more], [D:HP-IB#], 3, [RETURN].

Step 2. Exit command mode by pressing [COMMAND] again.

Verifying the "Destination" Assignment

The current "Destination" assignment(s) may be verified at any time by the *Show Assignments command.*

Example: Show the current "Destination" assignments.

Step 1. Press [COMMAND], [SHOW], [ASSIGN], [RETURN].

The following listing will appear on the LOG device (default is the display).

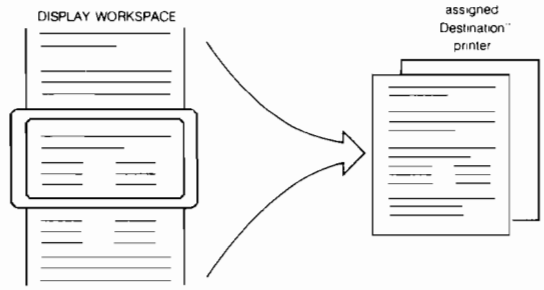
```
USER NAME     CURRENT ASSIGNMENTS
----------    -------------------
SOURCE        LEFT TAPE
DESTINATION   SHARED PRINTER#5
LOG           DISPLAY
PRINTER       EXTERNAL PRINTER
CTUL          CTUL???
CTUR          CTUR???
```

Recording Data from the Display

NOTE: The [RETURN] key may be pressed to stop the record operation.

Pressing [RECORD] copies data from the display to the current assigned "Destination" device; the cursor is "homed" and the entire contents of the current display workspace is copied to the destination.

Assign "Destination" to the printer:
Press COMMAND, ASSIGN, D:SH PTR <address> or
more, D:EX PTR or D:HP-IB# <address>, RETURN.
Press RECORD.



Recording Data to a Printer

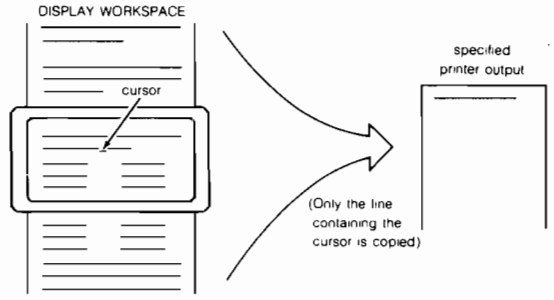
PRINTING ASCII DATA (USING THE COPY COMMAND)

Data may be copied from the display or diskette to the printer by the *Copy command.* Refer to "Data Operations" in Section 5 for detailed information on use of the Copy command.

From the Display

When copying from the display, remember that the copy starts at the line containing the cursor (i.e., the cursor need not be positioned in column 1; the entire line is copied anyway). If *COPY LINE* is specified, only the line containing the cursor is copied. If *COPY FILE* is specified, the copy operation starts with the line containing the cursor and stops at the bottom line of the display. If *COPY ALL* is specified, the copy operation stops at the end of the display workspace.

Press COMMAND, COPY, LINE, DISPLAY, EX PRNTR or
SH PRNTR <address> or HP-IB# <address>, RETURN.



Copying a Line from Display to a Printer

From a Disc File

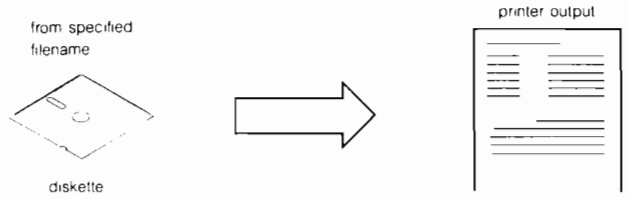
When copying data from a disc file either *Copy File* or *Copy All* will copy the entire contents of the file. File marks are converted to Form Feed characters and are sent to the printer. If *Copy Line* is specified, only the first line of the file will be copied. (Repeated *Copy Line* operations do not progress through the file. The *Copy Line command* is intended to be used with the tape emulator files.)

If you have two disc drives, specifying filenames on the second drive requires that the volumename or disc number be included: i.e.,

<filename>;<volumename or D#1 or D#2>

The semicolon, with no spaces, separates the filename from the volumename or disc number.

Press COMMAND, COPY, FILE or ALL, <disc filename>,
TO, EX PRNTR or SH PRNTR <address> or HP-IB# <address>,
RETURN.



Copying a File from a Diskette to a Printer

From a Tape Emulator File

When copying from a CTU emulator file, remember that the copy operation starts at the present position of the tape. (You may set the current tape position using the *Find File, Skip Lines,* or *Rewind command.*)

If *Copy Line* is specified, the copy starts at the beginning of the line (a record) on the file and stops at the beginning of the next line.

If *Copy File* is specified, the copy starts at the present position in the file and ends at the beginning of the next file. The file mark (which is converted to a Form Feed character, ᶠF) is sent to the printer.

If *Copy All* is specified, the copy starts at the present position in the file and ends at the end of the file. File marks and the end of the file are converted to Form Feed characters (ᶠF) and are sent to the printer.

PRINTING BINARY DATA (USING THE TRANSFER COMMAND)

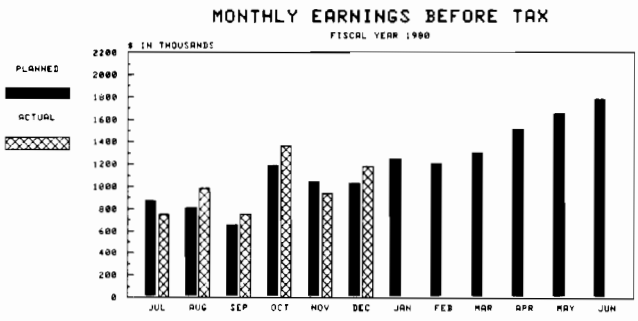
Binary data that you would copy to a printer is usually graphics image data. Refer to "Data Operations" in Section 5 for detailed information on use of the Transfer command.

The Transfer command copies 8-bit binary data; it is used in the same manner as the Copy command (i.e., a line, file, or all may be specified).

Example: Copy the graphics image data from graphics memory to the HP 7310A at address 8.

Step 1. Press next, next, TRANSFER, FILE, GRAPHICS, HP-IB, 8, RETURN .cf1

Step 2. Press COMMAND again to exit command mode.



Example of Graphics Image Data

PRINTER AS THE LOG DEVICE

The LOG device lists the data resulting from any of the *Show commands.* The default LOG device is the display, but a printer may be assigned by using the *Assign command.* The printer output provides a permanent record of the Show command listing should you require one.

Example: Assign the 2631A printer connected via the 13250B serial printer interface as the log device, and list the active and purged files on the diskette.

Step 1. Press COMMAND, ASSIGN, more, more, LOG, EX PRNTR, RETURN.

Step 2. Press SHOW, FILES, RETURN.

Step 3. Press SHOW, PURGE, RETURN.

```
Volume = X

  File        File    Start    Size       Create        Create
  Name        Type    Address  (sectors)  Date (M/D/Y)  Time
  ----------  -----   -------  ---------  ------------  --------
  TPM         Ascii       32        144     5/15/80     3:06 PM
  TPM1        Ascii      177        196     4/30/80     2:05 PM
  DEFINEMENU  Ascii      373          3     4/30/80     8:21 AM
  TPM2        Ascii      376        204     5/ 1/80     4:13 PM
  FDM         Ascii      580         74     5/15/80     1:10 PM
  USERSECT10  Ascii      656         36     5/ 5/80    10:21 AM
  SECT10A     Ascii      692         66     5/ 6/80     3:50 PM
  SECT10B     Ascii      758        118     5/ 7/80     9:01 AM
  CTUR001     Ascii      876          2     5/ 9/80    11:38 AM
  SECT10C     Ascii      878          0     5/ 9/80     9:42 AM
```

Example of Show Files

```
Volume = X

  File        File    Start    Size       Create        Create
  Name        Type    Address  (sectors)  Date (M/D/Y)  Time
  ----------  -----   -------  ---------  ------------  ------
  TPM1        Purged      176          1     4/30/80     2:05 PM
  USERSECT10  Purged      654          1     5/ 2/80    10:21 AM
  SECT10A     Purged      655          1     5/ 5/80     3:50 PM
```

Example of Show Purged Files

CONTROLLING THE PAPER

Skipping Lines

The *Skip Lines command* allows you to position the printer paper forward the number of lines specified in the command. (Actually the number of lines is converted to Line Feed (ᶫF) characters when sent to a printer.) Some printers (for example, the HP 9871A) permit you to skip backward by using a minus sign (−) before the number. Other printers ignore the minus sign and position the paper forward only.

Example: Skip forward ten lines on the HP 2631A/B Printer which is connected to the terminal via the 13250B Serial Printer Interface.

Step 1. Press [COMMAND], [next], [next], [SKIP], 10, [LINES], [EX PRNTR], [RETURN].

Step 2. Exit command mode by pressing [COMMAND] again.

Example: Skip forward four lines on the HP 7310A which is connected to the terminal via the 13296A Shared Peripheral Interface. (The address switches at the back of the printer have been set to 5.)

Step 1. Press [COMMAND], [next], [next], [SKIP] 4, [LINES], [HP-IB#], 5, [RETURN].

Step 2. Exit command mode by pressing [COMMAND] again.

Skipping Pages (Positioning to Top-of-Form)

The *Skip Page command* allows you to position the printer paper forward to the next top-of-form. (Actually, the command sends a Form Feed (FF) character to the printer.)

Example: Position the paper in the HP 2631A to the next top-of-form. (Assume that the printer is connected to the terminal via the 13238A Duplex Register.)

Step 1. Press [COMMAND], [next], [next], [SKIP], [PAGE], [EX PRNTR], [RETURN].

Step 2. Exit command mode by pressing [COMMAND] again.

The Effect of File Marks

File Marks that exist on tape emulator files on a diskettte will cause the printer to skip to top-of-form when a *Copy All command* to a printer is executed. (The file marks are converted to Form Feed characters (FF) when sent to a printer.) This is particularly useful when dumping forms data to a printer with preprinted forms.

The "Form Feed" Character

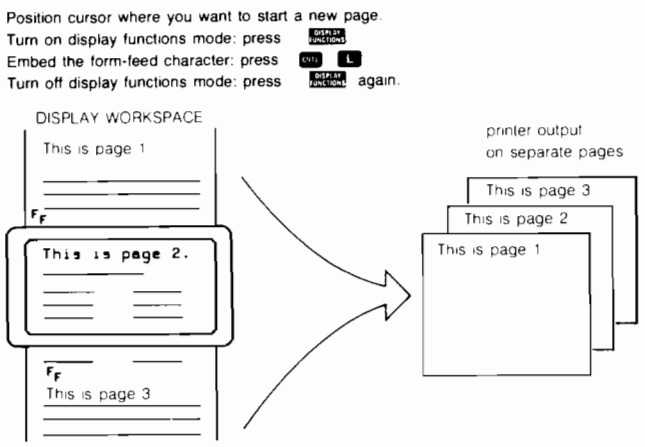
You may embed Form Feed characters in your text on the display by pressing [CNTL] [L] with Display Functions Mode on. This allows you to control the printed output so that only the text between the Form Feed characters is printed on each page.

Form Feed characters that are embedded on a disc file will be lost when read to the display; you may re-embed them using the same procedure.

The procedure is to position the cursor where you want the form feed to occur in the text, turn on display functions mode by pressing the [DISPLAY FUNCTIONS] key, press [CNTL] [L] to embed the Form Feed character, then turn off display functions mode by pressing the [DISPLAY FUNCTIONS] key again.

You can store the text with embedded Form Feed characters on a disc file, then dump the text to a printer at a later time, or you can copy the data directly from the display to a printer.

The Form Feed characters may be embedded after the text has been created by entering Edit Mode and using the procedure outlined above.



Embedding Form-Feed Characters (FF) in Your Text

COMPUTER OPERATIONS

SECTION 8

INTRODUCTION

This section provides operating instructions for communicating with a computer. Topics covered are: setting the terminal switches for data communication, autodialing (a feature which enables automatic dialing of a preselected telephone number by the terminal), sending data to the computer, and receiving data from the computer. Refer to the Data Communications section for detailed background information on data transfer between the terminal and a computer.

PREPARING THE TERMINAL FOR USE ON-LINE

After the terminal is installed and configured (refer to the Configuration section), very little need be done to use the terminal on-line. The proper interface PCA should be installed, and the modem or direct line connected (normally, this is already done). Make the settings listed in the following paragraphs.

Communication Switch Settings

There are three keyboard switches used to match the requirements of the computer that you will use. Additional information on communication settings is given in the Data Communiications section.

BAUD RATE. This switch sets the speed at which data will be transferred. If you do not know the correct speed, use 300 for operation with a modem or 2400 for hardwire operation (where the terminal is directly connected to a computer). If data from the computer appears garbled it may mean that you should select a different baud rate.

PARITY. This switch selects the type of data checking to be used. If the "■" character appears on the screen, you may have selected the wrong parity.

DUPLEX. This switch is normally set to FULL. If no characters are displayed when you type, switch to HALF.

Selecting Operating Modes

REMOTE. The [REMOTE] key must be down. This allows the terminal to send and receive data through its communication interface.

AUTO LF. Normally the [AUTO LF] key should be up.

CAPS LOOCK. Leave the [CAPS LOCK] key down unless the system you are using accepts lowercase letters.

BLOCK MODE. The [BLOCK MODE] key is used to select character-by-character or block transmission of data. When the key is up the terminal transmits characters to the computer as they are typed. This mode of operation is used for conventional exchanges with the computer system.

Example:

Computer `Please Type Your Company Name`

You type `HEWLETT-PACKARD` [RETURN]

Computer `What File Number Would You Like From The HEWLETT-PACKARD Library?`

You type `12345` [RETURN]

and so on . . .

When the [BLOCK MODE] key is down, characters are not sent as they are typed. Instead you can input information to the terminal and then edit it before sending it to the computer. The data is sent with the [ENTER] key. A portion of a line, a whole line, or even an entire screen of data can be sent at once. The size of the block of data sent depends on the setting of switches inside the terminal. (Refer to the Data Communications section for detailed information on the operation of Block Mode.)

Additional Settings

There are several more switch settings that can be made. These settings control such terminal features as line wraparound, the number of characters to be sent in block mode, and character overstrike (Space Overwrite Latch). Setting of these switches is normally required only when the terminal is first installed. These switches are located inside the terminal. Instructions for setting these switches are contained in the Configuration section.

If a modem is used it may be necessary to turn on the modem, make modem speed and parity settings, or dial a number. Baud rate, duplex, and parity settings should be the same values used for the terminal.

AUTODIALING (USA Only)

Your terminal can dial telephone numbers when it is used with the 13265A, option 001, 300 Baud Modem. The telephone numbers may be stored in the Assign table so that you can equate them to meaningful names when dialing.

```
USER NAME     CURRENT ASSIGNMENTS
-----------   -------------------
SOURCE        LEFT TAPE
DESTINATION   RIGHT TAPE
LOG           DISPLAY
PRINTER       EXTERNAL PRINTER
CTUL          CTUL
CTUR          CTUR
COMPUTER      735-8200
```

Before attempting to use the autodialing features, be sure that the 13260A Standard Asynchronous Communications Interface (02640-60239) is installed in the terminal. Refer to the *Modem User Manual*, part number 13265-90001, for modem operating instructions.) Make sure that the terminal's BAUD RATE switch on the keyboard is set to 300 and that the DUPLEX switch is set to FULL.

Using the DIAL Command

The *DIAL command* is used to execute the autodial feature.

Example: Dial 735-8200.

Step 1. Press [COMMAND], [next], [next], [next], [DIAL], 735-8200, [RETURN].

Example: Dial (408) 735-9685.

Step 1. Press [COMMAND], [next], [next], [next], [DIAL], (408) 735-9685, [RETURN].

Example: Dial the alphabetic characters equivalent to 735-7535, according to the telephone dial convention.

Step 1. Press [COMMAND], [next], [next], [next], [DIAL], PDL-PLDL, [RETURN].

Example: Assign the name "COMPUTER" to 735-8200, and dial the number using the assigned name.

Step 1. Press [COMMAND], [ASSIGN], [more], [more], [NAME], COMPUTER, [to], 735-8200, [RETURN].

Step 2. Verify the assign table entry by pressing: [SHOW], [ASSIGNS], [RETURN].

Step 3. Press [next], [next], [next], [DIAL], COMPUTER, [RETURN].

When the terminal begins to dial the number, the VOICE MODE indicator on the modem will turn off. The VOICE MODE and DATA MODE indicators will remain off while the number is being dialed. When dialing is completed, the DATA MODE indicator will turn on, and the following message will appear on the screen:

```
** Phone Connection Established **
```

At this point, press [COMMAND], [RETURN]. The computer system prompt will appear on the screen, and you may log on.

When you log off, the modem will automatically disconnect the terminal from the computer system. The VOICE MODE indicator on the modem will turn on. This telephone line is now freed for incoming or outgoing voice calls or for another autodialing session.

Using Parentheses and Hyphens

To represent the telephone number, you may use parentheses in the area code and a hyphen in the seven-digit number. However, these characters are not required. For example, you may enter a telephone number as "4087359693" instead of "(408) 735-9693".

Permissible Characters in the Assigned Name

The name you assign to the telephone number can be made up of the alphanumeric characters (A-Z), (a-z), and (0-9). The name may be 11 characters long; any characters over 11 will be truncated.

Using "@" for Slow Connecting Telephone Equipment

The "@" character is used as a wait mechanism for slow-connecting telephone equipment. Each "@" character causes a one-second delay. You may concatenate "@" characters together to increase the delay. They may be placed anywhere in the telephone number. For example, if you must dial "9" to get an outside line when calling from a business telephone, you would dial "9@735-7878".

Using ">" for Fast Dialing

The ">" character means to use the fast dialing rate instead of the slow dialing rate. This character should appear as the first character in the telephone number. The default dialing rate is slow, and prevails unless the ">" character is used.

SENDING DATA TO THE COMPUTER

Data can be sent to the computer from the terminal keyboard or from a disc file.

From the Keyboard

When not in Block Mode, data entered from the keyboard is sent immediately to the computer automatically. If you are in Block Mode (BLOCK MODE key down) then the data is stored in the terminal until you press the ENTER key. Block mode and the ENTER key are often used with data entry forms.

From a Disc File

Data can be read from a disc file and sent to the computer. This allows you to prepare data ahead of time, off-line, and then send the data later.

In Character Mode (BLOCK MODE key up), pressing READ causes the next file of data on the "Source" device (an assigned disc file) to be sent to the computer. The "Source" assignment may be displayed by pressing: COMMAND, SHOW, ASSIGNS, RETURN. It may be changed by the ASSIGN command sequence.

If the terminal is set for Block Mode (BLOCK MODE key down) the terminal will not send data to the computer when the READ key is pressed. In order to send data to the computer using the READ key when in Block Mode, you must use CNTL READ

READ **only**

Local:	Copy file to display
Remote:	
Character	Read file to computer (handshake per G, H straps)
Block	Copy file to display (allows editing before sending data to computer via the ENTER key)

CNTL READ

Local:	Copy file to display
Remote:	
Character	Read file to computer (no handshake)
Block, Line	Read file to computer (handshake per G, H straps)
Block, Page	Read file to computer (no handshake)

The Copy command can be used to send data to the computer, as follows: COMMAND, COPY, FILE, <filename>, to, <datacomm>, RETURN.

RECEIVING DATA FROM THE COMPUTER

To the Display

No special action is required to receive data from the computer. Data is normally displayed on the screen as it is received.

When operating at high data rates or when data is also being copied to a printer or other device it may be necessary to use "fill" or "pad" characters in the data to synchronize the terminal and computer. This will prevent possible data overruns. Detailed information on "fill" characters and other synchronization techniques are described in the Data Communications section.

To a Disc File

The computer can be programmed to send data directly to a disc file or you can direct data to a disc file from the keyboard by naming a disc file as a "Destination" device and then pressing the RECORD key before the computer begins sending the data.

A disc file may be assigned as the "Destination" at any time by pressing: COMMAND, ASSIGN, more, D:file, <filename>, RETURN.

The RECORD key should be pressed before the last response to the computer (typically before pressing RETURN) but before data is received from the computer. (See Data Logging.)

To A Printer

Data received from the computer can be routed to a printer in the same manner as described for the disc. Press: [COMMAND], [ASSIGN], [more], [D:EX PTR], [RETURN], [RECORD], [RETURN]. (See Data Logging.)

To Several Devices

You can route data received from the computer to several devices by using more than one device in the device assignment operation. For example: [COMMAND], [ASSIGN], [more], [D:file], <filename>, [EX PRNTR], [RETURN] will cause the data to be routed to both the disc file and the printer when the [RECORD] key is pressed.

To A Shared Device

Data from the computer can be directed to a shared device (such as printer with HP-IB device no. 4) by pressing: [COMMAND], [ASSIGN], [more], [D:HP-IB#], 4, [RETURN].

Data Logging

The terminal can be set to perform on-line data logging. This functions in a manner similar to Edit mode. Data received from the computer is displayed on the screen. When the display workspace is full, data at the top of the display workspace is routed to the "Destination" devices. It may be necessary to use "fill" characters to prevent data overruns when operating at high data rates. See the discussion of "fill" characters in the "To The Display" paragraph.

Data logging is performed by first selecting "Destination" devices, and then pressing: [COMMAND], [next], [next], [edit], [enable], [RETURN].

The indicator above the [f8] key will blink indicating that Data Logging Mode is on.

Data logging can only be done while the [REMOTE] key is down (On-Line operation). While performing data logging, the [REMOTE] key is disabled, and you cannot change "Source" or "Destination" device assignments.

When the data stream from the computer is finished, residual data will remain in display memory. To store this remaining data on the "Destination" devices, position the cursor in the line *following* the last line that you want copied; then press [RECORD]. The remaining data from the top of display memory to the line preceding the cursor will be stored on the destination devices. After the last line is transferred, the terminal beeps.

In order to return to normal operation after the data logging is complete, press: [COMMAND], [next], [next], [edit], [DISABLE], [RETURN].

The indicator above the [f8] key will turn off indicating that Data Logging Mode is off.

With the Mini Disc Drives

The display must be selected as the "Source" device, and a new disc file name must be selected as the "Destination" device. While in Remote mode, data logging mode is entered by pressing: [COMMAND], [next], [next], [edit], [ENABLE], [RETURN].

Data from the computer is written to the disc file via the display workspace. When all the data has been received, record the remaining data in the display workspace by pressing [RECORD]. Then exit Data Logging Mode by pressing: [COMMAND], [next], [next], [edit], [DISABLE], [RETURN].

Should the disc become full, the message:

```
No more disc space
```

will be displayed, data from the computer will be lost, and the keyboard will become inoperative except for the [RESET TERMINAL] key. Pressing the [RESET TERMINAL] key twice within a 1/2 second (a hard reset) will restore keyboard operation, and data from the computer will now be sent to the display. *You should anticipate the amount of data to be received and have the computer send only as much data as can be stored in the remaining disc space.* More diskettes could be used for storing a large amount of data.

Example: Assign "LOGDATA" as the destination disc file, and enter Data Logging Mode.

Step 1. Press [COMMAND], [ASSIGN], [more], [D:file], LOGDATA, [RETURN].

Step 2. Be sure that the [REMOTE] key is down, then press: [next], [next], [edit], [ENABLE], [RETURN].

Step 3. Press [COMMAND] again to exit command mode.

DATA COMMUNICATIONS

SECTION 9

INTRODUCTION

This section provides background information on data transfer between the terminal and a computer. The topics include means of connecting the terminal to the computer, data transmission modes (character mode and block mode), and data communication protocols. For information on installation procedures and cabling, refer to the Installation section. Complete data communications configuration information is contained in the Configuration section.

TERMINAL/COMPUTER CONNECTIONS

The terminal can be configured to work in a variety of computer applications. Your communication needs can be met by selecting a particular interface, modem, and protocol (communication control program).

Networks

The terminal can be connected in several network configurations. Figure 9-1 illustrates the following configurations:

- Hardwired to a computer (figure 9-1A).
- Connected to a computer through a modem (figure 9-1B).

Interface PCAs

The terminal can be used with one of two communication interface PCAs; the 13260A Standard Asynchronous Communications Interface PCA and the 13260B Extended Asynchronous Communications Interface PCA (see table 9-1). Table 9-2 lists the features of the 13260A and 13260B for easy comparison.

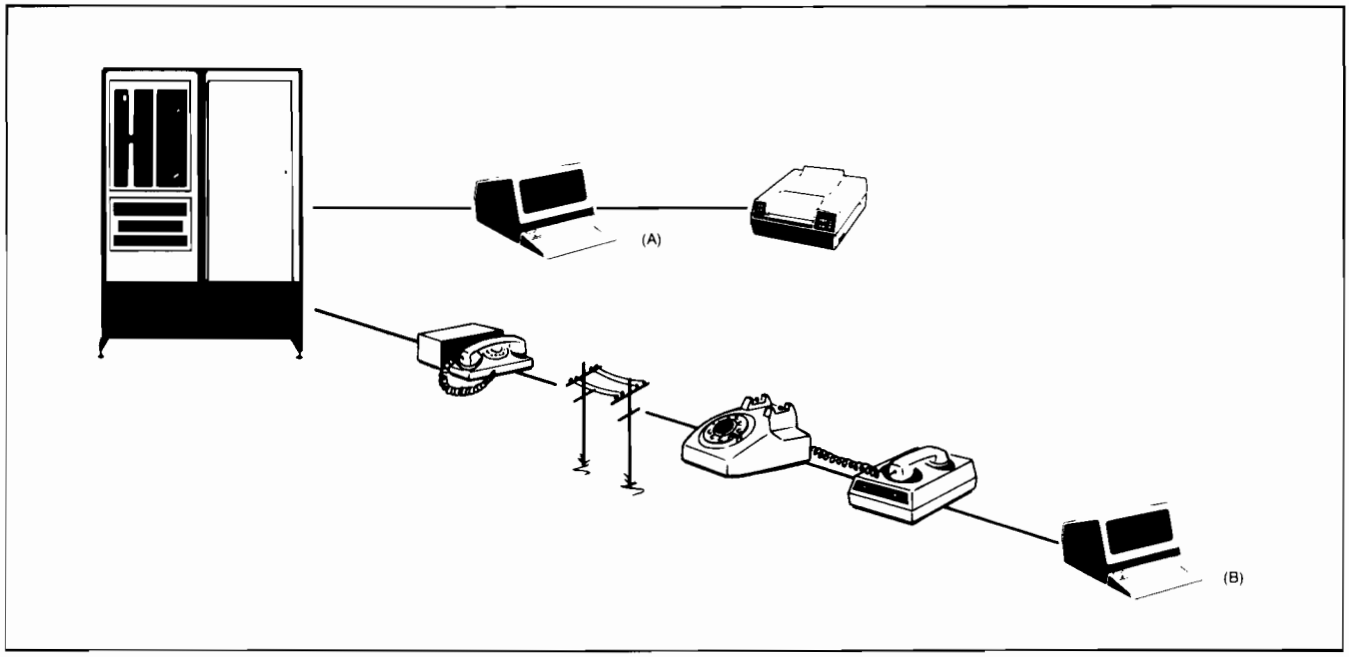


Figure 9-1. Terminal Network Configurations

Table 9-1. Data Communication Interface PCAs

Basic Communications (Point-to-Point)	
13260A (02640-60239)	Standard asynchronous communications interface (standard RS232C communications interface).
13260B (02640-60143)	Extended asynchronous communications interface (provides either standard RS-232C or 20 mA current loop communications. It allows split speed and custom baud rates).

Table 9-2. Data Communication Interface Capabilities

DATA COMMUNICATIONS FEATURES	13260 A	13260 B
Transfer Rate:		
110, 150, 300, 1200, 2400, 4800, 9600 bits per second and external clocking (110-9600)	X	X
Custom transfer rates within 1% from 37.5 to 2400 bits per second		X
Split speed transmit/receive capability		X
EIA RS232-C	X	X
Teletypewriter compatible	X	X
ASCII	X	X
20mA DC Current Loop	P	X
Transmission Modes:		
Character Transfer	X	X
Block Transfer	X	X
Half-duplex	X	X
Full-duplex	X	X
Asynchronous	X	X
Hardwired to computer; dialed (switched) or leased line	X	X
Modem Compatibility:		
Bell 103A, 202D, 202C, 202S, 202T (Asynchronous)	X	X
Vadic 3400 (Asynchronous/Synchronous)	X	X
HP 13265A	X	
Choice of main channel or reverse channel line turnaround for 202 modems	X	X
Auto-Answer/Disconnect	X	X
Transparency	X	X
Data Comm Self-Test	X	X
Error Checking:		
VRC, choice of parity generation/checking	X	X

P = Provides interface to HP 13266A Current Loop Converter

Once the interface has been selected, the terminal can be configured to operate with a variety of protocols, parities, and data formats. This is done by setting switches or jumpers on the datacomm interface PCA and the Processor PCA.

Interface Signals

The signals available on each of the communication interfaces are listed in the Installation section. This information can be used to verify interface compatibility or to fabricate special interface cables.

Modems

The terminal can be used with a variety of modems depending on the requirements of the given configuration or network. Table 9-3 contains a list of modems and the configurations in which they can be used.

DATA COMMUNICATION CONFIGURATION

Data communication configuration is done by setting switches on the keyboard, the Processor PCA, and the datacomm PCA. Refer to the Configuration section for configuration information.

COMMUNICATION PROTOCOLS

Control of computer-terminal communications is required for the orderly transfer of data. This control is provided in the form of a protocol or a set of rules and procedures. The protocol used determines who sends and who receives during each phase of communication. In addition the protocol normally provides for an orderly recovery from communication errors.

The various protocols can be selected by installing the proper interface and ROM modules. The terminal and the interface can then be configured to meet your specific requirements.

Table 9-3. Modems

MODEM	DATA RATE (BITS/SEC)	LINE TYPE: DIALED/LEASED	DUPLEX FULL/HALF	WIRES 2/4	REV. CHAN.
Asynchronous					
Bell 103A	300	D/L	H/F	2	No
Bell 202S Bell 202C ITT GH 2052 Nokia DS 9320	1200	D	H	2	Option
Bell 202T Bell 202D	1200 (3)	L	H/F	2/4	Option
Vadic VA3400	1200	D	F	2	No
Synchronous (1)					
Vadic VA3400	1200	D	F	2	No
Bell 201C Bell 201A Milgo 2200 Milgo 2400	2400	D/L (2)	H/F	2/4	No
Bell 208A	4800	L (2)	H/F	4	No
Bell 208B	4800	D	H	2	No
Bell 209A	9600	L (2) (4)	F	4	No

Notes:

1. Synchronous modems require the internal clock modem option.
2. Synchronous operation on a leased line requires the switched carrier modem option.
3. C2 line conditioning allows operation at 1800 bits/sec.
4. Requires D2 line conditioning.

The major characteristics of the available protocols are listed in table 9-4. The following paragraphs discuss each of these protocols.

Table 9-4. Protocol Characteristics

Single Terminal (Character Mode Protocols)	
Standard	Standard communication protocol is teletype compatible or can use the DC1 character to trigger multicharacter transfers.
Main Channel	Communication protocol uses special framing characters to control line turn-around.
Reverse Channel	Uses secondary channel signals to trigger line turn-around.

Character Protocols

Character protocols transmit a single character at a time. Data checking, if present, is done on individual characters only (parity). Some configurations allow the transmission of multicharacter groups but no block checks are made. There is no automatic retransmission of data following the detection of a data error. Currently available character mode protocols are Standard, Main Channel, and Reverse Channel.

Standard Communications is a term used to refer to *point-to-point* or single terminal communications. The terminal can be connected directly to a computer (hardwired) or through a modem. In most block applications the terminal can use a simple "handshake" protocol with the ASCII DC1 character. This protocol can be used with Bell 103 or equivalent modems (full-duplex operation). There are two additional protocols available, Main Channel and Reverse Channel. These protocols are normally only used with Bell 202 or equivalent modems (half-duplex operation).

Block Protocols

Block protocols transmit a block of characters at a time. Data checking is performed on an entire block of data. A separate block check character (BCC) is generated for each block. If a data error is detected, a retry of the data transmission is made automatically.

The remainder of this section provides descriptions and samples of control and data transfer sequences for various protocols.

DATA TRANSFER

The terminal can operate character-by-character as a completely interactive terminal or on a block of data at a time. Block transfers allow data to be composed and edited at the terminal allowing the user to verify and correct data before sending it to the computer.

Operating at High Speeds

If the number of characters sent to the terminal in one sequence exceeds 80, the required terminal processing time may cause some of the characters to be lost. (This usually does not occur at data rates of 4800 baud or less.) The symptom of this problem is the appearance of the "■" (delete) or "_" (underscore) characters.

There are two ways of insuring that this problem will not arise:

- It is possible to use a call-and-answer procedure between the terminal and the computer. If the computer sends an ENQ (octal 5) character after sending 80 characters, the terminal will respond with the ACK (octal 6) after it has processed the characters. The computer can then send the next block of characters. This is the recommended technique.

- Delays can be inserted in the application or system software after each 80 character transfer from the computer to the terminal. Transmitting NULL characters (octal 0) is one way to accomplish this. Each NULL character has the effect of 4 millisecond delay when operating at 2400 baud, and 2 milliseconds at 4800 baud. As an aid in calculating needed time delays, a list of processing times for various terminal functions is provided in table 9-5. The times listed are typical and can vary greatly depending on such factors as the number of characters in the terminal memory or on the display, and the current operating mode.

Table 9-5. Terminal Functions

TERMINAL FUNCTION	TYPICAL REQUIRED TIME (MILLISECONDS)*
Text Character	0.7
Cursor Up/Down/Left/Right	1.4
Line Feed	1.4
Insert Char	4.5
Delete Char	7.0
Insert Char w/wrap	12.0
Delete Char w/wrap	19.0
Soft Reset (Tapes Stationary)	130
Hard Reset (No Tapes)	800
Forms Mode On	12.0
Forms Mode:	
Home	8.0
Tab	8.0
Back Tab	10.0
Erase to End-of-line (40 characters)	10.0

*These times will increase if BASIC is active.

Character Mode

In Character Mode operation (BLOCK MODE key up) the terminal sends characters to the computer as they are typed. This mode of operation can be used for conversational exchanges with the computer.

Example:

Computer: `Please type your company name`

User types: `AJAX`

Computer: `What file number would you like from the AJAX library?`

User types: `12345`

and so on . . .

Multicharacter Transfers

There are certain functions that always result in multicharacter (block) data transfers.

- device input/output and control operations
- special function keys
- status requests
- cursor sensing
- all transfers while in Block Mode

In order for the terminal to make a block transfer, it must first be enabled and then triggered by the computer. Transfers are enabled by the ENTER or special function keys while the terminal is in Block Mode (see figure 9-2). When a transfer is enabled from the keyboard, the terminal sends a DC2 character to the computer to indicate that a data block is ready for transmission. (This process can be modified by strap settings on the Processor PCA). A transfer can also be enabled from the computer by an escape sequence requesting status (Esc^), cursor sensing (Esc a), or device control (Esc & p . . .) as shown in figure 9-3.

When the transfer is enabled the keyboard is locked out until the transfer is complete. Enabling sequences should not be entered from the keyboard or cartridge tapes because they will cause the keyboard to be locked until the computer responds with a DC1 character. (If the computer does not respond, a soft reset will cancel the transfer and re-enable the keyboard.)

Once a block transfer has been enabled, it must be triggered by the computer before the block of data is actually sent. The computer triggers the transfer by sending a DC1 character when it is ready to receive the data. The terminal also assumes that it has received the trigger when it is first powered up or fully reset, or when the REMOTE key is pressed (down).

The computer software must support the handshaking process used in multiple character transfers. The DC2 character must be recognized as a request to send data and the DC1 character must then be sent to trigger the transfer after buffers have been allocated to receive the data. Additional software support may be needed depending on your need for terminal or device control. There are straps on the Processor PCA that can be used to modify the handshaking process. These are discussed in the Configuration section.

NOTE: The computer should not be allowed to echo back information that has been transmitted as a block from the terminal.

Block Mode

When the terminal is in Block Mode (BLOCK MODE key down), characters are not transmitted as they are typed. Instead, the user can input data to the terminal, then edit and correct the data before sending it to the computer using the ENTER key. The data can be grouped into convenient blocks, either lines or pages. Block Mode operation allows you to efficiently utilize computer and communication facilities.

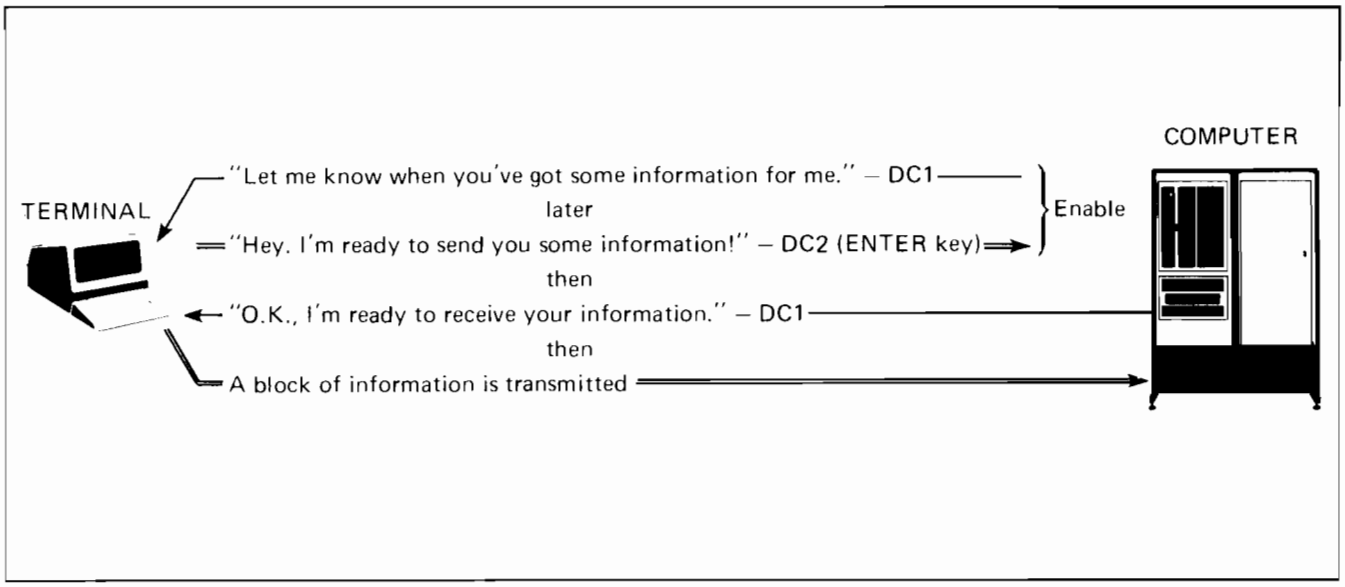


Figure 9-2. Block Transfer Enabled by the ENTER Key

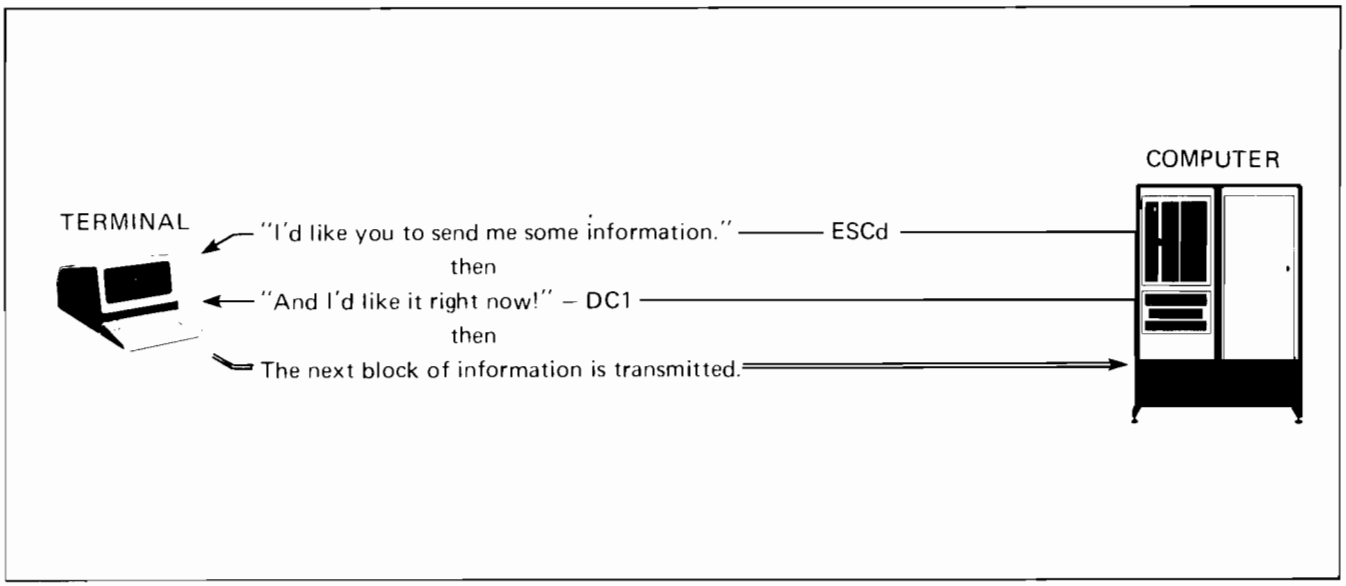


Figure 9-3. Block Transfer Enabled by the Computer

The G and H switches on the Processor PCA are used to control the terminal's response to block transfer requests (refer to the Configuration section for complete Processor PCA configuration information).

SWITCH G	SETTING H	BLOCK OPERATION
Closed	Closed	Data transfers used DC1/DC2 handshake. Other transfers are triggered by the receipt of a DC1 character.
Closed	Closed	Data is sent when the ENTER key is pressed. Other block transfers are triggered by the receipt of a DC1 character.
Open	Closed	All block transfers require a DC1/DC2 handshake.
Open	Open	No DC1/DC2 handshake is required for any block transfer.

Note: In half duplex operation, a line turnaround is substituted for a DC1 character.

NOTE: In half duplex operation, a line turnaround is substituted for a DC1 character.

The size of the block of information transferred in Block mode, and the control characters used to separate fields and to terminate blocks differ somewhat, depending on the Line/Page strapping of the terminal (on the Processor PCA) and whether or not the terminal is operating in FORMAT MODE. Figure 9-4 illustrates these differences..

In the example in figure 9-5, the user has an application in which order data is to be entered in the same format as a standard company form.

Full Duplex Operation

In full duplex operation, the characters which are typed at the keyboard are transmitted to the computer and are not displayed unless they are returned by the computer. This setting is ignored when in Block Mode.

Teletype Compatible Communications

In teletype compatible (full duplex, character mode) applications, the terminal can be quickly configured for use by following the instructions given in the Configuration section. Note that if block data transfers are used the computer should be programmed to use the simple DC1/DC2 protocol described under Multicharacter Transfers earlier in this section.

Half Duplex Operation (202 Modem Compatibility)

In half duplex operation, data is sent in only one direction at a time. In order to change the direction of data flow, a line turn around must occur. This means that the sender becomes the receiver and the receiver becomes the sender. Line turn arounds are controlled by half duplex line protocols. Both the computer and the terminal must use the same protocol otherwise malfunction and loss of data will result. The Main Channel and Reverse Channel protocols are examples of half duplex operation.

Initially the terminal is in the transmit state. While in this state the terminal will ignore data sent from the computer. The terminal will remain in the transmit state until one or more of the following occur:

- An ON to OFF transition on the SB (CCITT 122) line (Reverse Channel)
- An end of data character (ETX or EOT) is sent (Main Channel)
- The user tries to send an end of data character from the keyboard (control-C, control-D).

The above conditions cause the terminal to switch to the receive state.

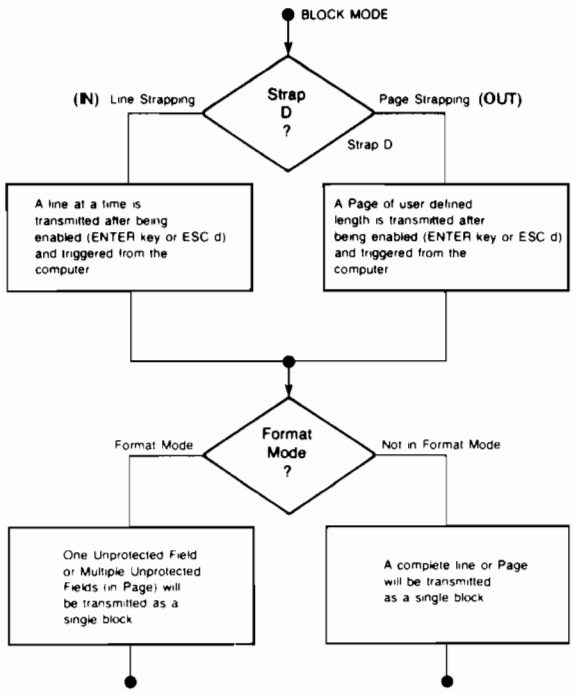
The terminal then receives the processed data until one of the following occurs:

- An ON to OFF transition of the CF (CCITT 109) line (Reverse Channel)
- An end of data character (ETX or EOT) is received (Main Channel)

The terminal then requests the computer or modem for permission to transmit. The computer or modem responds with transitions of the CB (CCITT 106) and SB (CCITT 122) lines. (If the computer or modem does not respond within 2.6 seconds the terminal will return to the receive state.) If the computer is ready the terminal will begin to send any data present in its output buffer.

The terminal provides a range of half duplex line protocols, including Bell 202 modem compatible protocols. These protocols are selected by switch settings on the Processor PCA.

Half-duplex operation can be controlled either by RS232C signal lines or by control characters in the data being transferred or by a combination of characters and signals. The Main Channel protocol uses control characters while the Reverse Channel protocol uses control signal lines.

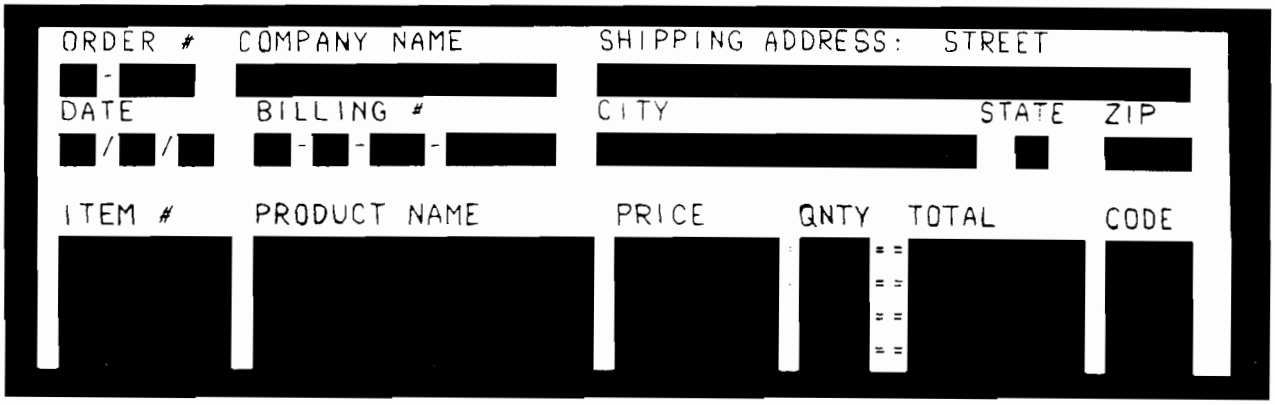


	STRAPPED FOR LINE	STRAPPED FOR PAGE
non-FORMAT MODE	• data is transferred from the current cursor position to the end of the line or to a Record Separator (RS) control character, whichever occurs first. • imbedded control characters are transmitted. If present, the RS character is sent. • the Block is terminated by the transmission of a CR(LF), a Carriage Return and Line Feed if AUTO LF is depressed. (A local CR(LF) is executed to reposition the cursor; if no more information is present at or beyond the cursor the transmission consists of RS CR(LF).)	• data is transferred from the current cursor position to the end of the terminal's allocated memory or to the next RS, whichever occurs first. Thus the Block to be transferred could be several lines of information. • imbedded control characters are transmitted. If present, the RS character is sent. • if multiple lines are in the Block, they are separated by CR LF in the transfer. The Block is terminated by the transmission of an RS.
FORMAT MODE	• only information in Unprotected Fields is transmitted. If the cursor is not in an Unprotected Field it will be forwarded to the next one or RS CR(LF) will be transmitted if no such field exists. Data is transmitted from the cursor position to the end of the Field or an RS, whichever occurs first. Thus the Unprotected Field to be transferred could no be longer than one line in length. • imbedded display control characters are not transmitted. If present, the RS character is sent. • the Block is terminated by the transmission of a CR(LF) and the cursor is forwarded one character position.	• only information in Unprotected Fields is transmitted. If the cursor is not in an Unprotected Field it will be forwarded to the next one or RS will be transmitted if no such fields exist. Data found in Unprotected Fields is transmitted from the cursor until an RS or the end of memory is encountered. • imbedded display control characters are not transmitted. If present the RS character is sent. • a Unit Separator (US) control character (or RS character for multipoint) is transmitted between each Unprotected or Transmit Only field. The Block is terminated by the transmission of an RS.

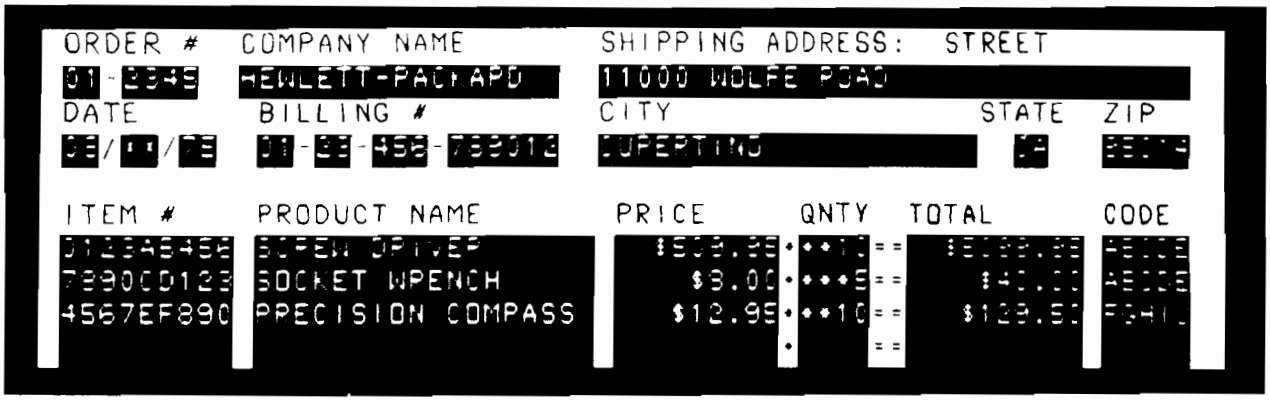
Figure 9-4. Block Mode Operation

STEP 1. The user presses the Special Function key, which he has previously programmed in a remote computer routine to both automatically display the form shown and turn on FORMAT MODE. (REMOTE and BLOCK MODE are depressed.)

STEP 2. All areas of the display have been programmed to be protected except for the dark fields within the form itself. Thus, as data is typed at the keyboard only these dark areas can be written into. The cursor automatically will tab from one field to the next when a field boundary is encountered or by use of the TAB key. The user now inputs data from the keyboard.



The complete form would look as follows:



STEP 3. After filling out the form and correcting any noticed errors, the ENTER key is pressed once. The following sequence of events would then occur:

- Having received a DC1 from the computer, the terminal transmits a DC2.
- Computer software recognizes the DC2 and responds with a second DC1.
- The terminal receives DC1 and transmits all data as one Block, fields separated by US's and the Block terminated by an RS.

STEP 4. The form full of data has been transmitted to the computer. The user could then Home the cursor, hit CLEAR DISPLAY to clear only the data from the form in FORMAT MODE, and enter a second set of data inputs — repeating the sequence and reusing the form.

Figure 9-5. Example of Format Mode with Page Strapping

MAIN CHANNEL (CHARACTER CONTROL) PROTOCOL. The Main Channel protocol is for use in half-duplex or Bell 202 modem equivalent networks where secondary channel signals are not available. The Main Channel protocol uses control characters to "frame" each data transmission. These framing characters indicate to the receiving station that a data transmission has begun or ended.

An ASCII STX (octal 002) character can be used to indicate the start of a data transmission. An ASCII ETX (octal 003) or EOT (octal 004) character is used to indicate the end of a data transmission. When these characters are received they are used to perform a line turn-around.

The following switch settings should be made on the Processor PCA to operate using the Main Channel protocol (refer to the Configuration section for complete configuration of the Processor PCA).

SWITCH	SETTING	DESCRIPTION
R	Open	
S,T	Closed,Open Open,Closed Open,Open	<STX>data<ETX> data<EOT> data<ETX>

Note that at least one of the S or T switches must be open to select Main Channel protocol.

Example:

U,V,W,X,Y,Z — All Open - Variations of the Main Channel Protocol are discussed under Other Protocols.

The operation of the Main Channel protocol is shown in figure 9-6. Sample data transfers are shown in figure 9-7. Figures 9-7a and 9-7b illustrate the line turn-arounds that occur during a log-on sequence when in block mode. Figures 9-7c and 9-7d illustrate the transfers that occur during character mode operation.

REVERSE CHANNEL (SIGNAL LINE CONTROL) PROTOCOL. The Reverse Channel protocol is for use in half-duplex or Bell 202 modem equivalent networks where secondary channel signals are available. The Reverse Channel protocol uses changes on secondary channel lines SA (CCITT 120) and SB (CCITT 122) to control line turn-arounds.

The following settings should be made on the Processor PCA to operate using the Reverse Channel protocol:

SWITCH	SETTING	DESCRIPTION
R	Closed	Monitor the CB line
S,T	Closed,Closed	Reverse Channel (No framing characters)
U	Closed	Watch for computer interrupts (SB>0)
V	Closed	Watch for Carrier (CF) transitions

Example:

W,X,Y,Z — All Open - Variations of the Reverse Channel protocol are discussed under Other Protocols.

The operation of the Reverse Channel protocol is shown in figure 9-8. Sample data transfers are shown in figure 9-9.

OTHER PROTOCOLS. In addition to the Main and Reverse Channel protocols you can select various features of both the configure a custom protocol to suit your own requirements. A flowchart of the overall Basic Communications function including the Half-Duplex settings is given in the appendix. You can create a custom protocol using this flowchart and the switch descriptions in table 9-8.

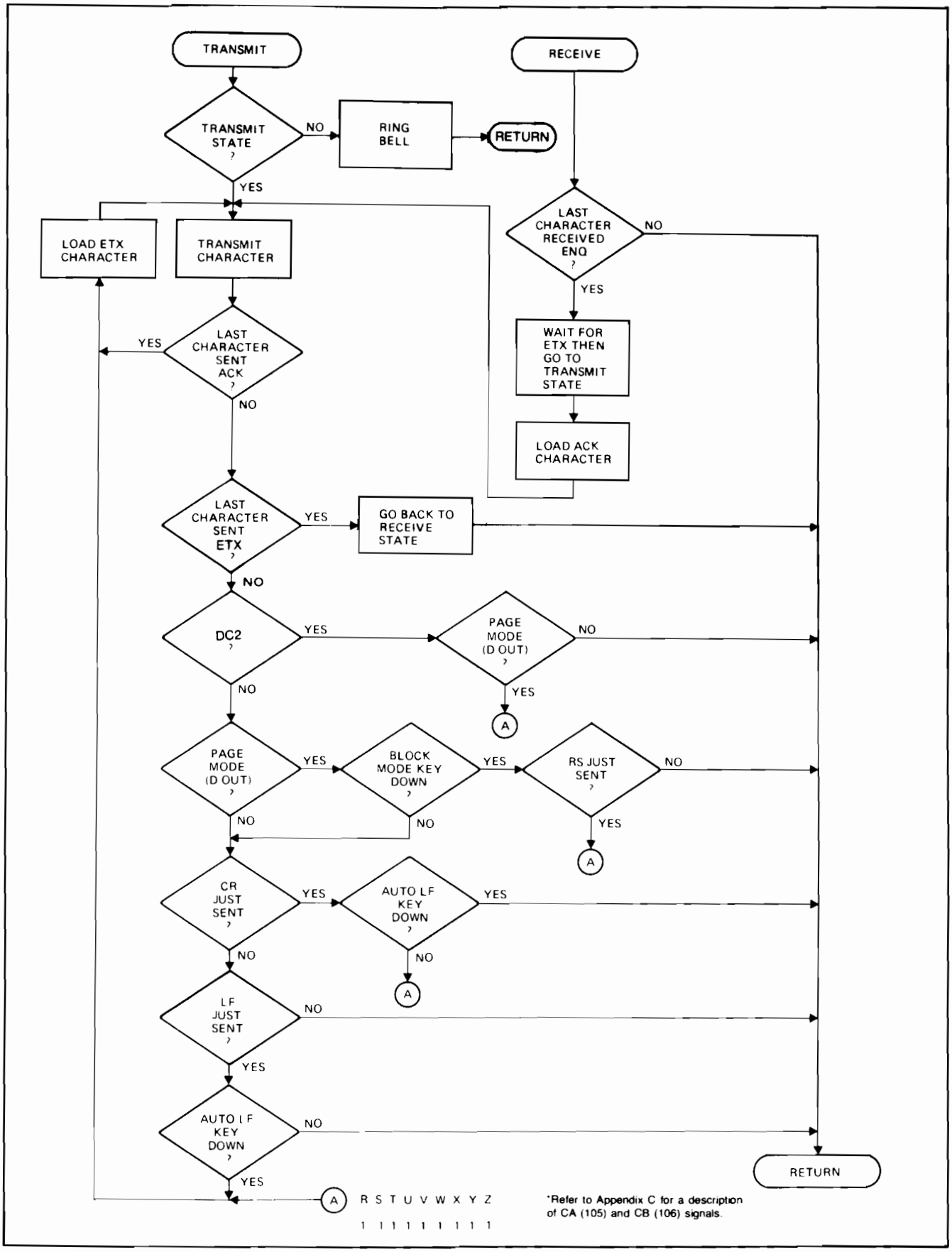


Figure 9-6. Main Channel Protocol

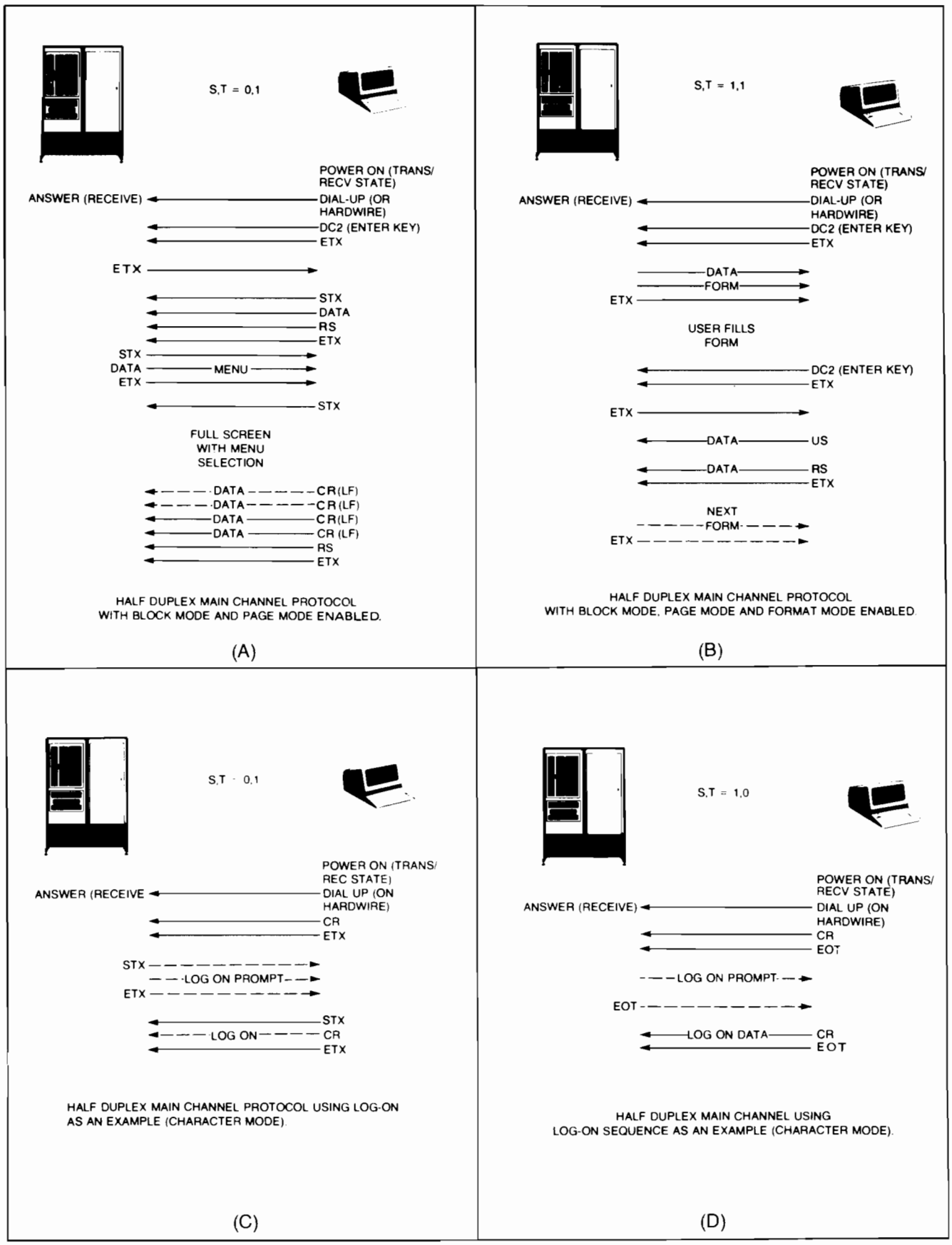


Figure 9-7. Sample Data Transfers Using Main Channel Protocol

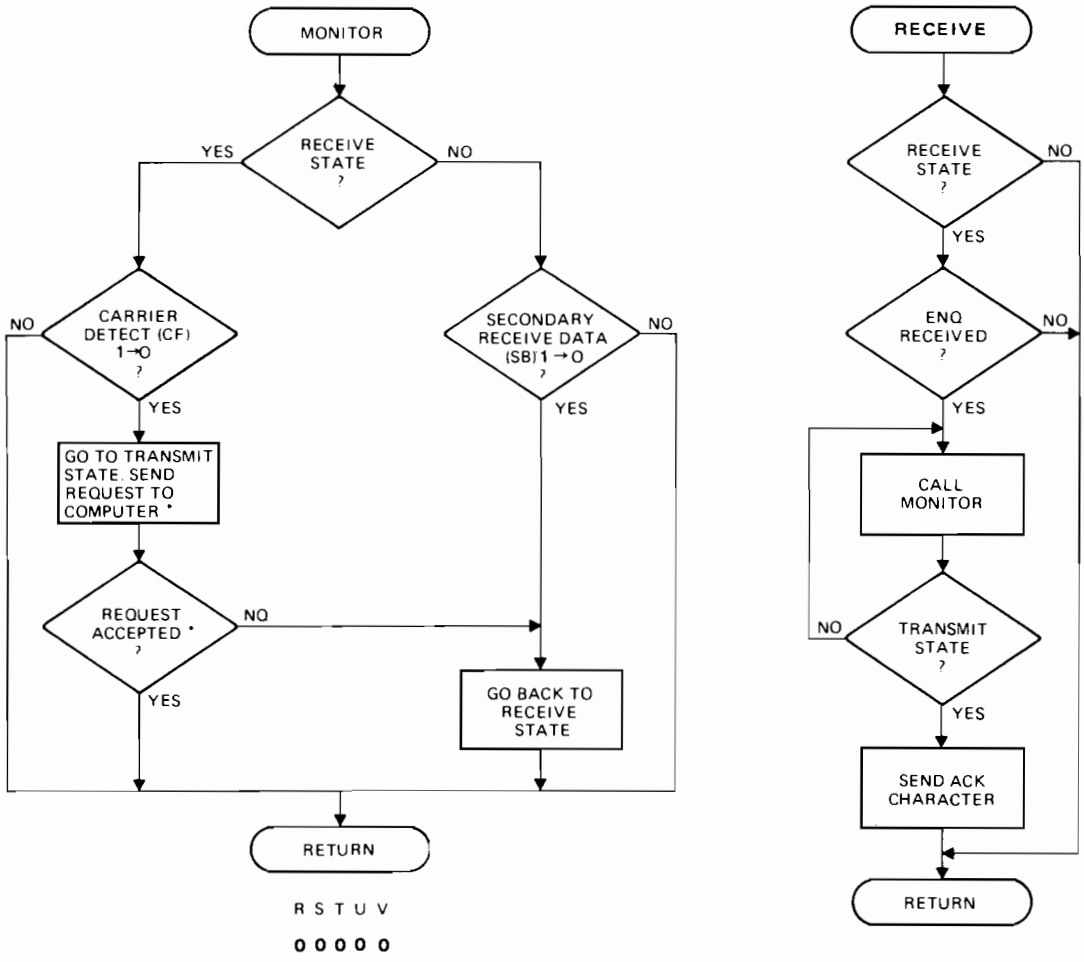


Figure 9-8. Reverse Channel Protocol

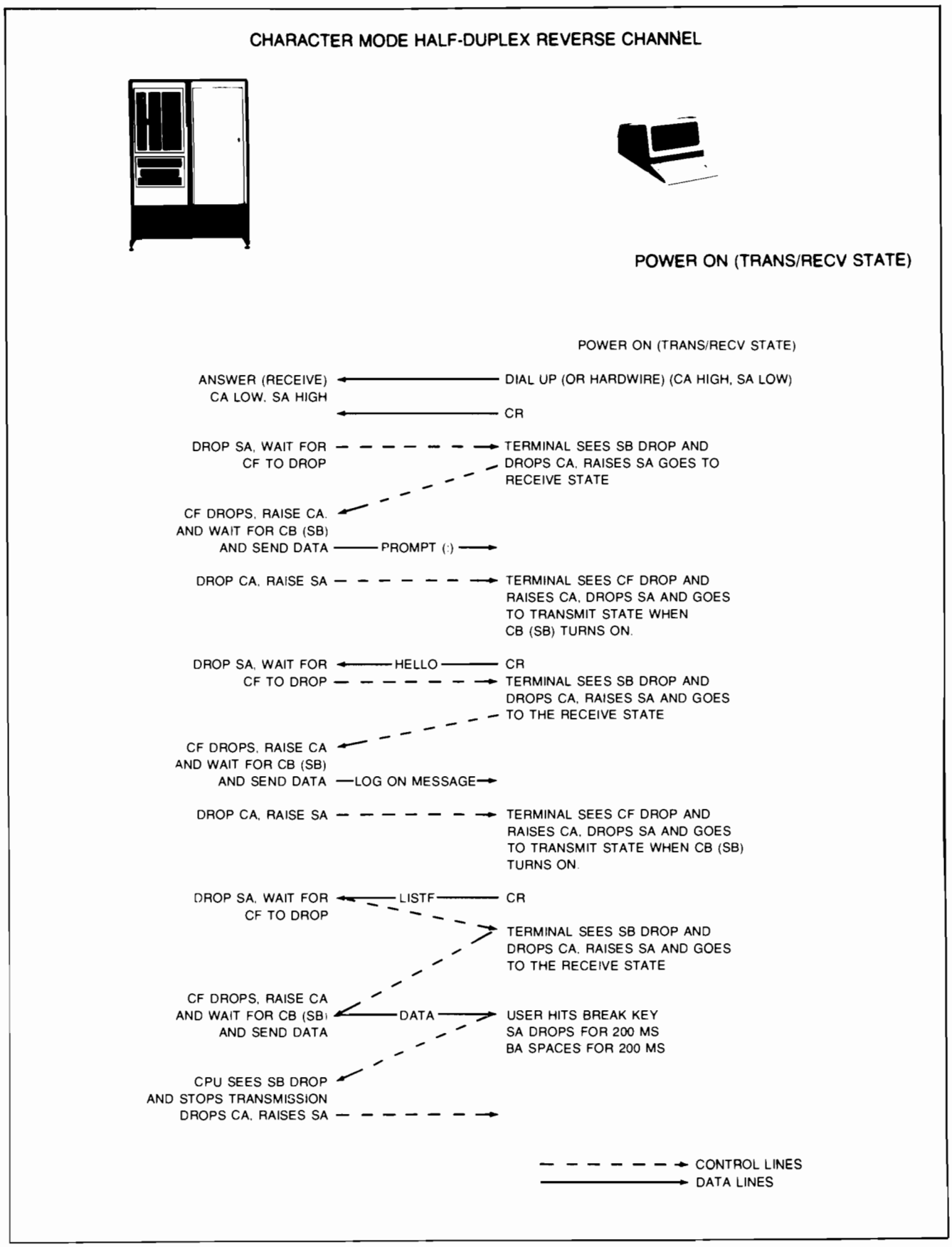


Figure 9-9. Sample Data Transfers Using Reverse Channel Protocol

STATUS

SECTION 10

INTRODUCTION

This section contains information on how to obtain and interpret terminal status information. In addition to terminal status, you can also obtain status information on input/output devices used with the terminal, the terminal's identification and memory size, the status of any command sent to the terminal, and graphics status.

Status requests are made by sending an escape code sequence to the terminal to select the desired status information. All status requests are treated as block transfers. (Refer to Multicharacter Transfers in Section 9.) The examples that follow use the DC1 character to trigger the status transfer (Basic Communication Protocol). Only one status request at a time can be enabled. The last status block requested will be returned when the DC1 is received.

TERMINAL STATUS

In response to status requests the terminal returns an escape code sequence followed by one or more bytes. The status bytes are followed by a terminator. The terminator received may be a ␛ (↳), or ␛ depending on the communications protocol and terminal configuration (refer to sections 9 and 11). The examples that follow use the ␛ character as a terminator.

The status information is normally contained in the lower four bits of each status byte. The upper five bits of the bytes are set so that the byte will have the value of an ASCII character. Each byte can be interpreted as one of 32 characters as shown in table 10-1.

Terminal status is made up of 14 status bytes (bytes 0-13) containing information such as display memory size, switch settings, Processor PCA configuration, and terminal errors. These fourteen status bytes are displayed below the terminal Self-Test pattern when the TEST key is pressed. There are two terminal status requests, primary and secondary. Each returns a set of 7 status bytes.

Primary Terminal Status

The first block of terminal status (bytes 0–6) is requested by sending the following escape sequence:

Primary Terminal Status Request: ␛^

The terminal will respond with an ␛\ and 7 status bytes followed by a terminator. A typical primary terminal status request and response is shown in figure 10-1. The example is for a configuration requiring the DC1 character to trigger block transfers.

Table 10-1. ASCII Status Characters

ASCII CHARACTER	BINARY	ASCII CHARACTER	BINARY
SPACE	0010 0000	0	0011 0000
!	0010 0001	1	0011 0001
"	0010 0010	2	0011 0010
#	0010 0011	3	0011 0011
$	0010 0100	4	0011 0100
%	0010 0101	5	0011 0101
&	0010 0110	6	0011 0110
'	0010 0111	7	0011 0111
(	0010 1000	8	0011 1000
)	0010 1001	9	0011 1001
*	0010 1010	:	0011 1010
+	0010 1011	;	0011 1011
,	0010 1100	<	0011 1100
-	0010 1101	=	0011 1101
.	0010 1110	>	0011 1110
/	0010 1111	?	0011 1111

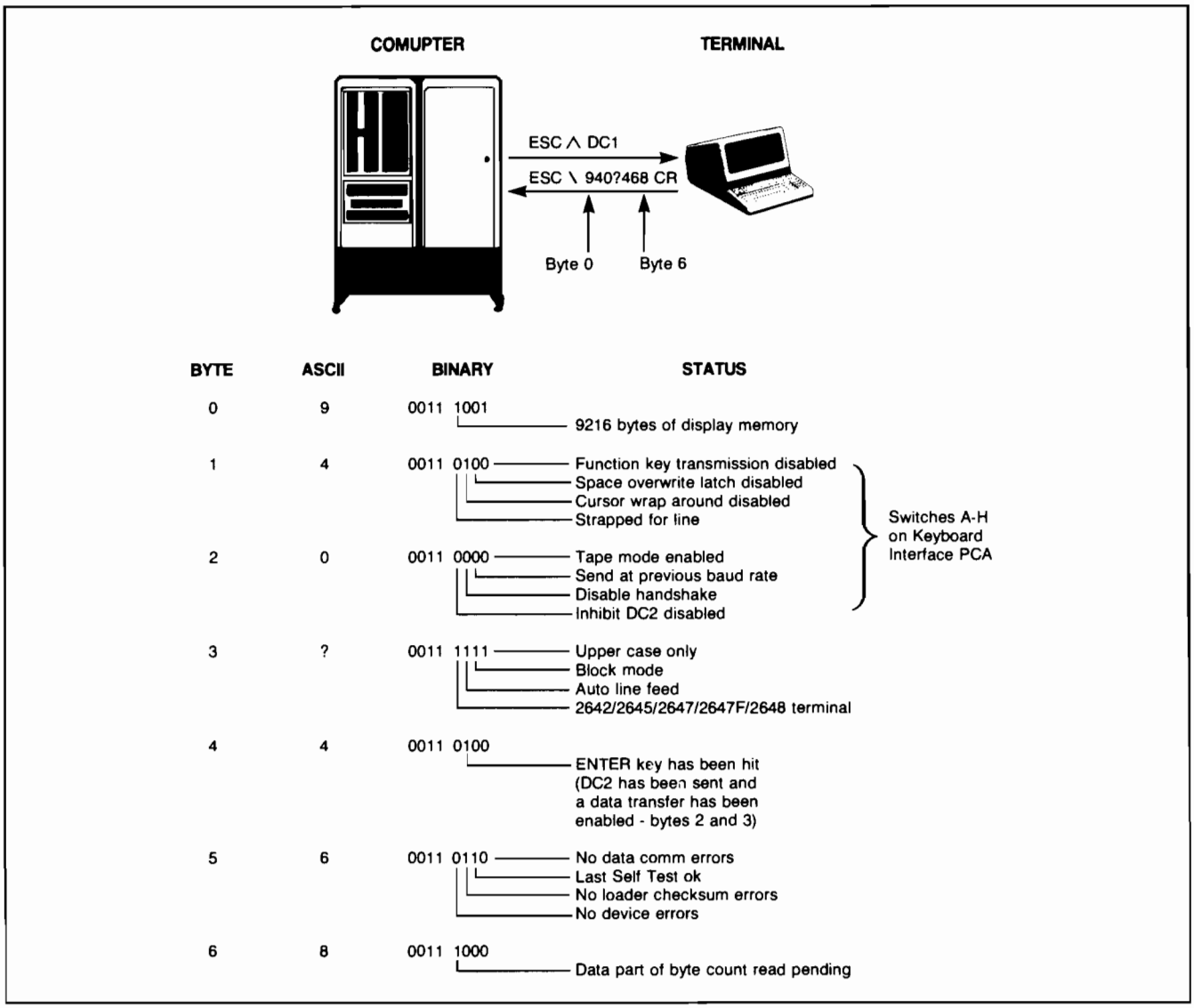


Figure 10-1. Primary Terminal Status Example

PRIMARY TERMINAL STATUS

BYTE 0 DISPLAY MEMORY SIZE

8	7	6	5	4	3	2	1
0	0	1	1	1/0	1/0	1/0	1/0

1/0 = 1024, 2048, 4096, 8192

The amount of display memory (blocks of 1K) available in the terminal is returned. The amount can range from 1K to 12K bytes.

BYTE 1 PROCESSOR PCA SWITCHES (A-D)

8	7	6	5	4	3	2	1
0	0	1	1	1/0	1/0	1/0	1/0

- Switch D — Page/Line: 1 = open (Page); 0 = closed (Line)
- Switch C — (End-of-Line Wraparound): 1 = open (Disabled); 0 = closed (Enabled)
- Switch A — (Function Key Transmission): 1 = open (Transmitted); 0 = closed (Not transmitted)
- Switch B — (Space Overwrite Latch): 1 = open (Enabled); 0 = closed (Disabled)

BYTE 2 PROCESSOR PCA SWITCHES (E-H)

8	7	6	5	4	3	2	1
0	0	1	1	1/0	1/0	1/0	1/0

- Switch H (Inhibit DC2): 1 = open (Enabled); 0 = closed (Disabled)
- Switch G (DC2 Handshake): 1 = open (Enabled); 0 = closed (Disabled)
- Switch E (Tape Mode): 1 = open (Disabled); 0 = closed (Enabled)
- Switch F (Fast Binary Read): 1 = open (Switch to 9600 baud); 0 = closed (Retain speed)

BYTE 3 LATCHING KEYS

8	7	6	5	4	3	2	1
0	0	1	1	1	1/0	1/0	1/0

- terminal type (2642/2645/2647/2647F/2648)
- AUTO LF Key: 1 = auto LF; 0 = no auto LF
- CAPS LOCK Key: 1 = upper case only; 0 = upper and lower case
- BLOCK MODE Key: 1 = block mode; 0 = character mode

BYTE 4 TRANSFER PENDING FLAGS

8	7	6	5	4	3	2	1
0	0	1	1	1/0	1/0	1/0	1/0

- Secondary Status Pending: 1 = yes; 0 = no
- ENTER Key Pending: 1 = yes; 0 = no
- Cursor Sense Pending: 1 = yes; 0 = yes
- Function Key Pending: 1 = yes; 0 = no

BYTE 5 ERROR FLAGS

8	7	6	5	4	3	2	1
0	0	1	1	1/0	1/0	1/0	1/0

- Device Error: 1 = error; 0 = no error
- Normally 0. After a binary load operation: 1 = Successful binary load; 0 = Unsuccessful binary load
- Data Comm: 1 = parity or buffer overflow error; 0 = no error
- Normally 0. After a self-test: 1 = successful self-test; 0 = error in self-test.

BYTE 6 DEVICE TRANSFER PENDING FLAGS

8	7	6	5	4	3	2	1
0	0	1	1	1/0	1/0	1/0	1/0

- Data Portion of Byte Count Read Pending: 1 = yes; 0 = no
- I/O Read Pending: 1 = yes; 0 = no
- Device Status Pending: 1 = yes; 0 = no
- Device Completion Pending: 1 = yes; 0 = no

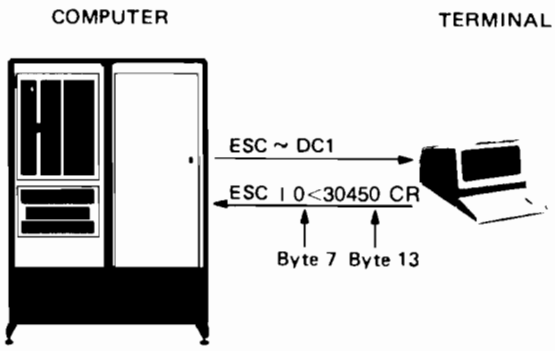
Figure 10-2. Interpretation of Primary Terminal Status

Secondary Terminal Status

The second block of terminal status (bytes 7-13) is requested by sending the following escape sequence:

Secondary Terminal Status Request: ESC ~

The terminal will respond with an ESC I and 7 status bytes followed by a terminator. A typical secondary terminal status request and response are shown in figure 10-3.



BYTE	ASCII	BINARY	STATUS
7	0	0000	No buffer memory installed
8	<	1100	I/O firmware not installed APL firmware not installed Graphics firmware installed Programmable terminal
9	3	0011	Auto terminate enabled Clear terminator enabled Data Comm Self Test enabled Normal operation of INSERT/DELETE CHAR — Switches J-M on Processor PCA
10	0	0000	Switches N-R closed
11	4	0100	Switch S closed Switch T closed Switch U open Switch V closed
12	5	0101	Switch W open Switch X closed Switch Y open Switch Z closed
13	0	0000	Memory not locked

Figure 10-3. Secondary Terminal Status Example

SECONDARY STATUS BYTES

BYTE 7 BUFFER MEMORY

8	7	6	5	4	3	2	1
0	0	1	1	0	1/0	0	0

1 = 4096 bytes
0 = none

Memory installed in addition to display memory that is available for use as data buffers. (The terminal always has 4K for buffers.)

BYTE 8 TERMINAL FIRMWARE CONFIGURATION

8	7	6	5	4	3	2	1
0	0	1	1	1	1	0	1/0

1 = Programmable terminal
0 = Terminal not programmable

1 = Graphics firmware installed
0 = No Graphics firmware

1 = I/O firmware installed
0 = not installed

1 = APL Firmware
0 = No APL Firmware

The device support firmware is required before printers can be used with the terminal.

BYTE 9 PROCESSOR PCA SWITCHES (J-M)

8	7	6	5	4	3	2	1
0	0	1	1	1/0	1/0	1/0	1/0

Switch M (Alternate Operation - INSERT AND DELETE CHARACTER Keys)
1 = open (Invert wrap sense)
0 = closed (normal)

Switch L (Self Test Inhibit)
1 = open (Inhibit test)
0 = closed (Allow test)

Switch J (Auto Terminate)
1 = open (Enabled)
0 = closed (Disabled)

Switch K (Clear Terminator)
1 = open (Enabled)
0 = closed (Disabled)

BYTE 10 PROCESSOR PCA KEYS (N-R)

8	7	6	5	4	3	2	1
0	0	1	1	1/0	1/0	1/0	1/0

Switch R (varies with communications protocol)
1 = open
0 = closed

Switch Q Compatibility Mode (Unscaled)
1 = Enabled
0 = Disabled

Switch N Printer (Escape Code Transfer)
1 = open (Send ESC code)
0 = closed (Do not send code)

Switch P Compatibility Mode (Scaled)
1 = Enabled
0 = Disabled

Note that if either or both of the P or Q switches is enabled an extended data comm buffer is selected.

BYTE 11 PROCESSOR PCA KEYS (S-V)

8	7	6	5	4	3	2	1
0	0	1	1	1/0	1/0	1/0	1/0

Switch V
1 = open
0 = closed

Switch U
1 = open
0 = closed

Switch S
1 = open
0 = closed

Switch T
1 = open
0 = closed

The use switches S to V varies depending on the communication protocol used.

BYTE 12 PROCESSOR PCA SWITCHES (W-Z)

8	7	6	5	4	3	2	1
0	0	1	1	1/0	1/0	1/0	1/0

Switch Z (Parity)
1 = (Force Parity)
0 = (Do not Force Parity)

Switch Y (Transmit light)
1 = open (On when CC high)
0 = closed (On when CB high)

Switch W (Data Comm Test)
1 = open (Inhibit)
0 = closed (Allow)

Switch X (Speed Select)
1 = open (CH = ON)
0 = closed (CH = OFF)

The use switches W to Z varies depending on the communication protocol used.

BYTE 13 MEMORY LOCK/BI-LINGUAL MODE

8	7	6	5	4	3	2	1
0	0	1	1	1/0	1/0	1/0	1/0

1 = APL Mode
0 = ASCII Mode

1 = locked in row ≠ 0
0 = locked in row 0

1 = memory lock on
0 = memory lock off

1 = memory full
0 = memory not full

Figure 10-4. Interpretation of Secondary Terminal Status

DEVICE STATUS

The status of a printer can be obtained by a device status request. This request would typically be made following an input/output operation or as a result of testing bytes 5 and 6 of the terminal status.

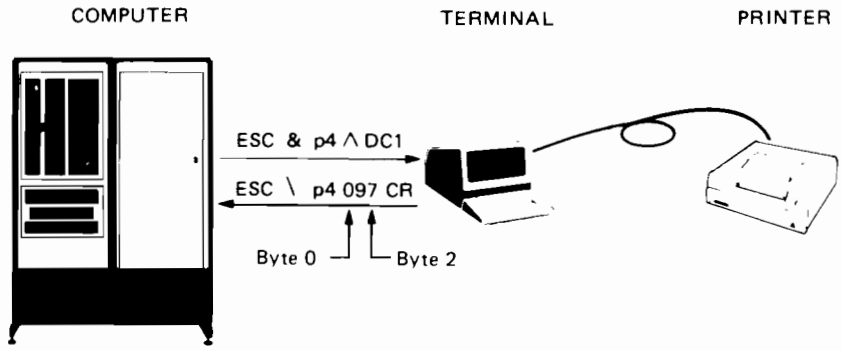
Device status is requested by sending the following escape sequence:

Device Status Request: `ESC & p <device> ^`

where: `<device>` is 1, 2, or 4 and
1 = left tape
2 = right tape
4 = printer

NOTE: The printer designated by device code 4 is usually an 8-bit duplex or serial printer. To equate an HP-IB printer connected by HP-IB bus to device code 4, the Assign command must equate "PRINTER" to the address of the HP-IB printer. (Refer to the Assign command in the Commands section.)

The terminal will return an `ESC \p <device code>` and 3 bytes of device status followed by a terminator. A typical device status request and response are shown in figure 10-5. A status request from device 3 (display) will be ignored.



BYTE	ASCII	BINARY	STATUS
0	0	0011 0000	No print error Paper present
1	9	0011 1001	Printer busy Last print command executed properly
2	7	0011 0111	Printer is on-line Baud rate is 300 characters per second

Figure 10-5. Device Status Example

TAPE UNITS

BYTE 0

8	7	6	5	4	3	2	1
0	0	1	1	1/0	1/0	1/0	1/0

End of File
1 = at end of file (tape positioned after the file mark)
0 = not at end of file

Load Point
1 = at load point
0 = not at load point

Write Error (Write Backspace/Read Mode only)
1 = error
0 = no error

End of Tape
1 = at end of tape
0 = not at end of tape

BYTE 1

8	7	6	5	4	3	2	1
0	0	1	1	1/0	1/0	1/0	1/0

Command Execution
1 last command performed
0 last command aborted

Write Protect
1 protected
0 not protected

Tape Busy
1 busy*
0 not busy

Read Error
1 error during last read
0 no error

A busy indication is returned when the terminal is

conditioning the tape
rewinding the tape
finding a file (keyboard or cartridge tape initiated)
skipping lines (keyboard or cartridge tape initiated)
no tape present

Since the terminal cannot process a status request while performing a normal read or write operation, these functions will not result in a busy indication

BYTE 2

8	7	6	5	4	3	2	1
0	0	1	1	1/0	1/0	1/0	1/0

Soft Error (read/write error-recovered)
1 yes
0 no

Hard Error (10 read/write failures)
1 yes
0 no

Tape Inserted
1 yes
0 no

End of Valid Data
1 yes
0 no

PRINTERS

BYTE 0

8	7	6	5	4	3	2	1
0	0	1	1	0	0	1/0	1/0

Paper Out (varies with printer)
1 = yes
0 = no

Print Error (varies with printer)
1 = yes
0 = no

BYTE 1

8	7	6	5	4	3	2	1
0	0	1	1	1/0	0	0	1/0

Command Execution
1 last command performed
0 last command aborted

Printer Busy
1 = yes
0 = no

BYTE 2

8	7	6	5	4	3	2	1
0	0	1	1	1/0	1/0	1/0	1/0

Printer Baud Rate

Printer Connected
1 = yes
0 = no

rate	bit 4 3 2
external	0 0 0
110	0 0 1
150	0 1 0
300	0 1 1
1200	1 0 0
2400	1 0 1
4800	1 1 0
9600	1 1 1

Figure 10-6. Interpretation of Device Status

SELF IDENTIFICATION STATUS

In addition to terminal status you can request self identification and memory size status information. The self identification request is initiated by sending an ESC * s followed by a single parameter (1) followed by a ^.

If the DC1 handshake is enabled (Processor PCA straps G and H), the status block is not actually sent until receipt of a DC1 character. If the DC1 character is used, only one status request can be enabled while the terminal is waiting for a DC1. When the DC1 is received, the last status block requested will be sent.

The Processor PCA straps determine the terminating characters sent following the status block (CR/LF or ETX). Status requests turn on an echo suppress mode in the terminal. This prevents information echoed back from the computer from being displayed on the screen. Once a status block has been sent, characters received by the terminal will not be displayed until one of the following control characters is received:

NUL, CR, ESC, ETX, EOT, LF, ENQ, ACK, BEL

With the exception of CR and LF the terminating control code itself will be executed.

The terminal expects the status information to be echoed and uses the terminating control character to turn off the suppress echo mode. If the computer does not echo the status back, a suitable control character must be returned to the terminal to turn off the echo suppress mode.

Read Device I.D.

When you request a device I.D. the terminal responds with its Hewlett-Packard model number, 2647F.

Device I.D. Request: ESC * s 1 ^

The terminal responds: 2647F <terminator>

Memory Size Request

The memory size on the specified bank is returned. This memory starts at 0 and is assumed to be no greater than 48K:

Memory Size: ESC & b <bank id> p^ or ESC & c <bank id> p^

where <bank id> = 0, 1, 2, or 3

The terminal responds:

<number of bytes of memory> <terminator>

COMMAND STATUS

A program executing in a host computer can obtain command execution error messages from the terminal by sending the following escape sequence:

Command Status Request: ESC,c REPORT STATUS OF COMMAND CR

In response the terminal returns a five-digit decimal value. A value of 00000 indicates that no error occurred. To extract the error code from a non-zero value, you convert the value of a 16-bit binary number, as follows:

Decimal Value: 00309
Hexadecimal Value: 0135
16-bit Binary Representation:

Bit:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0	0	0	0	0	0	0	1	0	0	1	1	0	1	0	1
Value:					1 0 2 4	5 1 2	2 5 6		1 2 8	6 4	3 2	1 6	8	4	2	1

Refer to the appendix (Error Messages) for conversion of the error code to an error message.

GRAPHICS STATUS

In addition to terminal status you can request graphics status information. All graphics status requests are initiated by sending an ESC * s followed by a single parameter (1 through 13) followed by a ^. The single parameter selects the particular status block desired. If an invalid parameter is used, the terminal will simply return its I.D. (see Read Device I.D.).

Graphics Status Request: ESC * s <parameter> ^

where: ESC * s is the graphics status escape sequence.

<parameter> is 1-12 and selects one of twelve blocks of graphics status information.

The graphics status blocks that can be requested are listed in table 10-2 together with the format of the terminal's response. Detailed descriptions of each of the status requests are contained in the following paragraphs.

The terminal will respond with one or more bytes of status information followed by a block terminator. All status information is returned in ASCII format, separated by commas. Coordinates are returned in a fixed format, consisting of a sign and five digits. Leading zeros are used as required to provide a fixed number of digits (i.e. +00100, −01234). This allows you to use simple input statements without the need to mask or shift bits.

Table 10-2. Graphics Status Requests

PARAMETER	REQUEST	RESPONSE
1	Read device I.D.	2647F
2	Read current pen position	<X>,<Y>,<PEN>
3	Read graphics cursor position	<X>,<Y>
4	Read graphics cursor position with wait	<X>,<Y>,<KEY>
5	Read display size	<LLX>,<LLY>,<URX>,<URY>,<MMX>,<MMY>
6	Read device capabilities	<b1>,<b2>,...<b15>,<b16>
7	Read graphics text status	<X size>,<Y size>,<origin>,<angle>,<slant>
8	Read zoom status	<size>,<ON/OFF>
9	Read relocatable origin	<X>,<Y>
10	Read Reset status	<RESET>,<b1>...<b6>,<b7>
11	Read area shading capability	1,8,8
12	Read Graphics Modification capabilities	1,1

If the DC1 handshake protocol is enabled (processor PCA straps G and H), the status block is not actually sent until receipt of a DC1 character. If the DC1 character is used, only one status request can be enabled while the terminal is waiting for a DC1. When the DC1 is received, the last status block requested will be sent.

The Processor PCA straps determine the terminating characters sent following the status block (␍/␊ or ␍). Graphic status requests turn on an echo suppress mode in the terminal. This prevents information echoed back from the computer from being displayed on the screen. Once a status block has been sent, characters received by the terminal will not be displayed until one of the following control characters is received:

␀, ␈, ␅, ␉, ␋, ␊, ␌, ␎, ␏

With the exception of ␈ and ␊ the terminating control code itself will be executed.

The terminal expects the status information to be echoed and uses the terminating control character to turn off the suppress echo mode. If the computer does not echo the status back, a suitable control character must be returned to the terminal to turn off the echo suppress mode.

The graphics status blocks that can be requested are:

Read Device I.D. (Parameter=1)

When you request a device I.D. the terminal responds with its Hewlett-Packard model number, 2647F.

Device I.D. Request: ␅ * s 1 ^

The terminal responds: 2647F <terminator>

Read Current Pen Position (Parameter=2)

The pen position and status are returned as a string of ASCII characters.

Pen Position Request: ␅ * s 2 ^

The terminal responds: <X>,<Y>,<Pen>,<terminator>

where: <X> = X coordinate
<Y> = Y coordinate
<Pen> = Pen state, 0=pen up, 1=pen down

For example, assume that the pen is at 360, 80, the pen is up, and the terminal is set for the DC1 handshake, with CR as the terminator:

The computer sends: ␅ * s 2 ^ <terminator> DC1

The terminal responds: +00360,+00080,0␍

X coordinate Pen state

Y coordinate

Read Graphics Cursor Position (Parameter=3)

The graphics cursor position is returned as a string of ASCII characters.

Read Graphics Cursor Request: `Esc * s 3 ^`

The terminal responds: `<X>,<Y> <terminator>`

where: `<X>` = X coordinate
`<Y>` = Y coordinate

Read Cursor Position with Wait (Parameter=4)

This request allows the user to position the cursor, then strike a key to return the position. The ASCII decimal code for the key struck is also returned (not the actual character). The code is returned as three digits. For example, striking an uppercase A would return 065. Only ASCII character keys will generate a response (i.e. ROLL UP, ROLL DOWN, etc. are ignored). The graphics cursor is turned on, if not already on. If an escape sequence is received by the terminal after it has received the READ CURSOR with WAIT command and before a key is struck, the READ CURSOR command will be aborted. The new sequence will be executed instead.

Read Graphics Cursor with Wait Request: `Esc * s 4 ^`

The terminal responds: `<X>,<Y>,<key code> <terminator>`

where: `<X>` = X coordinate
`<Y>` = Y coordinate
`<key code>` = Decimal value of key struck

The position bytes are ordered as in the read pen request.

Read Display Size (Parameter=5)

This request returns the number of displayable units in the X and Y axes. It also returns the number of units per millimeter in the display. This request allows you to scale data for use on graphic devices with varying display areas.

Read Display Size Request: `Esc * s 5 ^`

The terminal responds: `<LLX>,<LLY>,<URX>,<URY>, <MMX>,<MMY><terminator>`

where: `<LLX>,<URX>` = Lower left and upper right x coordinates
`<LLY>,<URY>` = Lower left and upper right y coordinates
`<MMX>,<MMY>` = number of units per millimeter in the x and y axes, (five digits and a decimal point)

The terminal will always return a fixed response. The lower left corner has coordinates of 0,0. The upper right corner has coordinates of 719,359. There are approximately 3 units per millimeter in each axis.

Terminal response: `+00000,+00000,+00719,+00359, 00003.,00003.<terminator>`

Read Device Capabilities (Parameter=6)

The device capabilities request returns a list of graphic and plotting features available in the terminal. This allows you to use one program for a variety of graphic devices. Not all of the features listed are available in the terminal. The absence of a feature is indicated by a 0. If a feature is present, it may be necessary to send an additional request to determine the exact capabilities present. Where multiple response values are possible the terminal's standard response is shaded.

Device Capability Request: `Esc * s 6 ^`

The terminal responds:
`<b1>,<b2>,<b3>,<b4>,<b5>,<b6>,<b7>, <b8>,<b9>,<b10>,<b11>,<b12>,<b13>,<b14>, <b15>,<b16><terminator>`

where:

`<b1>` = Clear Display
0 = no clear
1 = paper advance
2 = clear (total erase)
3 = partial clear by area

`<b2>` = Number of Pens (1)

`<b3>,<b4>` = Not Used (0,0)

`<b5>` = Area Shading
0 = no
1 = yes (see Read Area Shading Capability)

<b6>,<b7> = Not Used (0,0)

<b8> = Dynamic Modification
0 = no
1 = yes (see Read Modification Capability)

<b9> = Graphics Character Size
0 = fixed
1 = integer multiples of the basic cell size
2 = any size

<b10> = Graphics Character Angles
0 = fixed
1 = multiples of 90°
2 = multiples of 45°
3 = any angle

<b11> = Graphics Character Slant
0 = fixed
1 = 45°
2 = any angle

<b12> = Dot-Dash Line Patterns
0 = none
1 = predefined only
2 = user defined and predefined

<b13>-<b16> = Not Used (0,0,0,0)

The terminal will always respond:

3,1,0,0,1,0,0,1,1,1,1,2,0,0,0,0<terminator>

Read Graphics Text Status (Parameter=7)

The terminal returns the current text size, orientation, slant, and type of justification. Refer to the Graphics Control section, for a description of graphics text characteristics.

Read Graphics Text Request: Et * s 7 ^

The terminal returns: <x size>,<Y size>,<origin>,<angle>,<slant><terminator>

where: <X size> = X dimension of the character cell (three digits)
<Y size> = Y dimension of the character cell (three digits)
<origin> = Relative position of text to cursor (see text origin command)
<angle> = Text angle 0, 90, 180, or 270 (five digits and a decimal point)
<slant> = 00000. or 00045. degrees

Read Zoom Status (Parameter=8)

This request returns the terminal's zoom setting.

Read Zoom Status Request: Et * s 8 ^

The terminal responds:

<zoom size>,<zoom on/off><terminator>

where: <zoom size> = zoom setting, 1-16 (three digits and a decimal point)
<zoom on/off> = 0 for Off, 1 for On

Read Relocatable Origin (Parameter=9)

The position of the relocatable origin is returned as x and y coordinates.

Read Relocatable Origin Request: Et * s 9 ^

The terminal responds:

<X coordinate>,<Y coordinate><terminator>

Read Reset Status (Parameter=10)

You can determine whether or not the terminal has executed a full reset (or Power On) since the last time reset status was checked. This will tell you whether or not you need to reestablish terminal settings or images before resuming terminal functions. An additional seven bytes are returned but are not used.

Read Reset Status Request: Et * s 10 ^

The terminal responds: <reset>,<b1>,<b2>,<b3>,<b4>,<b5>,<b6>,<b7><terminator>

where: <reset status> = 0 No full reset since last check
or
1 Terminal has been reset
<b1>-<b7> = 0 (not used)

Read Area Shading Capability (Parameter=11)

The area shading capability of the terminal can be read. These are fixed for the terminal.

Read Area Shading Request: `ESC * s 11 ^`

The terminal will always respond: `1,8,8 <terminator>`

The "1" indicates that the area shaded must be rectangular. The first "8" indicates that the shading pattern is 8 units wide. The second "8" indicates that the shading pattern is 8 units high.

Read Graphics Modification Capabilities (Parameter=12)

You can read the terminal's dynamic graphics capabilities. This is the ability of the terminal to change selected portions of the display. These are fixed for the terminal.

Read Graphic Modification Capabilities Request: `ESC * s 12 ^`

The terminal will always respond: `1,1 <terminator>`

These two bytes indicate that the terminal has selective erase and compliment capabilities.

Any Other Parameter

Any other parameter which has not been assigned causes the terminal I.D. to be returned. This is to prevent an invalid status request from tying up the requesting computer while waiting for a response.

`2647F <terminator>`

Byte:	<byte 1>	<byte 2>	<byte 3>	<byte 142>	<byte 144>
Bit:	5 4 3 2 1	5 4 3 2 1	5 4 3 2 1	5 4 3 2 1	5 4 3 2 1
Data:	1 1 0 0 0	1 1 1 0 0	0 0 0 0 0	0 0 1 1 1	0 0 0 1 1
Display:	* *	* * *		* * *	* *

INSTALLATION

SECTION 11

INTRODUCTION

This section contains installation instructions for the terminal. Also included are instructions for selecting ac operating voltages (115 or 230V), selecting optional operating functions, and installing terminal add-on accessories.

WARNING

Hazardous voltages are present inside equipment. The procedures contained in this section shall be performed only by qualified service personnel.

VORSICHT

Innerhalb des Gerats bestehen gefahrliche Spannungen. Die in diesem Abschnitt enthaltenen Arbeiten durfen nur durch Betriebsfachpersonal durchgefuhrt werden.

ATTENTION

Des tensions dangereuses sont presentes a l'interieur du materiel. Les operations decrites dans cette section ne devront etre effectuees que par un personnel qualifie.

AVVISO

Pericolo: Alta tensione presente in questa apparecchiatura. Le procedure contenute in questa sezione debbono essere effettuate soltanto da qualificato personale di servizio.

ADVERTENCIA

Hay voltaje peligroso en el interior de este equipo. Los procedimientos expuestos en esta seccion solo debera llevarlos a cabo el personal de servicio calificado.

高圧危険

内部装置に危険な高電圧がかかっています この章にある処置や手順に関しては、専門のサービスマンによってのみ行なって下さい

OPENING THE TERMINAL

To gain access to the terminal internal components, open the terminal as follows (also see figure 11-1):

1. Set mainframe rear panel power switch to off and disconnect power cord from power connector.

NOTE: Mainframe top cover is unlocked by inserting access key supplied with terminal in each of the keyways located on right and left sides of top cover. Inserting keys into keyways unlock top cover. No key rotation is required.

2. From front of terminal, insert access key into right keyway and unlock right side of terminal by slightly raising right side of top cover (figure 11-1, A and B).

3. While maintaining upward pressure to keep right side of terminal unlocked, insert access key into left keyway and raise top cover until both right and left sides of terminal are unlocked (figure 11-1, C).

4. Using both hands, carefully swing top cover up until it latches into the half open position (figure 11-1, D).

NOTE: The half open position provides adequate room for performing most service routines. However, if extensive repairs are to be made or if components contained in the top cover are to be serviced, fully open mainframe in accordance with step "e".

CAUTION

Mainframe top hinges are open hinge type. When fully opening terminals do not allow top hinges to slip off hinge pins.

5. Firmly grasp top cover in one hand and release safety latch (see figure 11-2) by pressing it inboard with other hand. Then, using both hands, swing top cover up and over to a full open position (resting on its top).

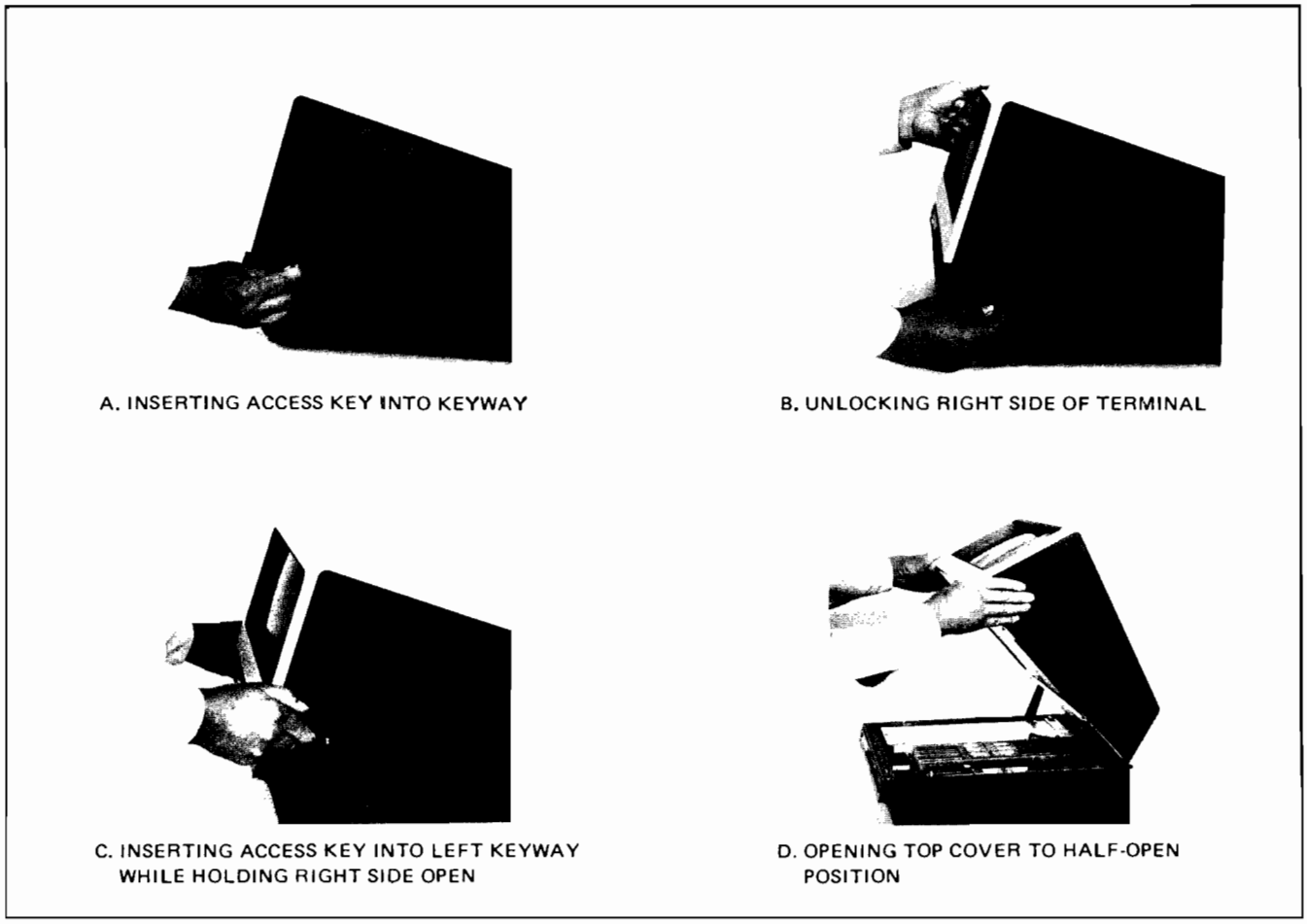


Figure 11-1. Opening the Terminal

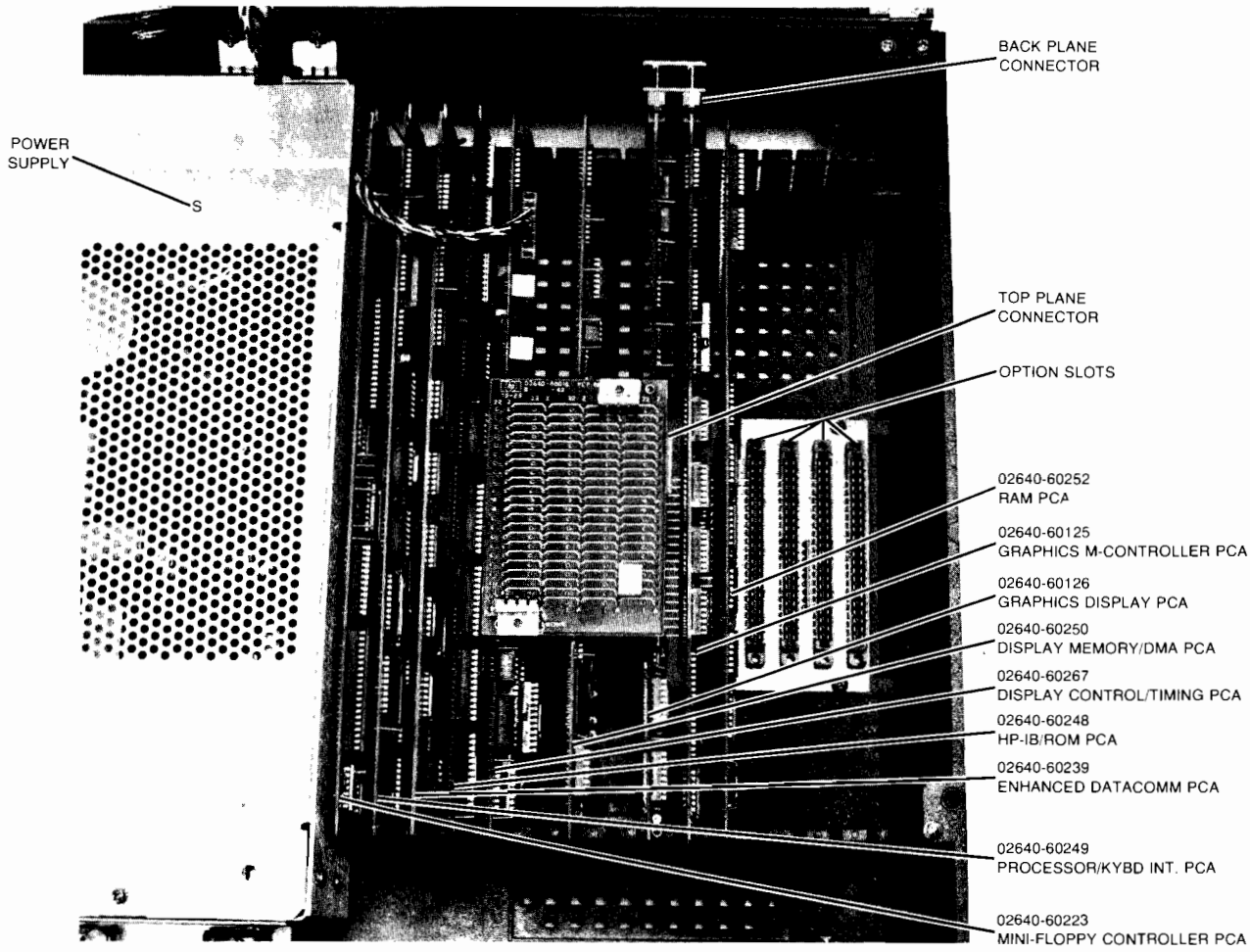


Figure 11-2. Mainframe Part Locations (Sheet 1 of 2)

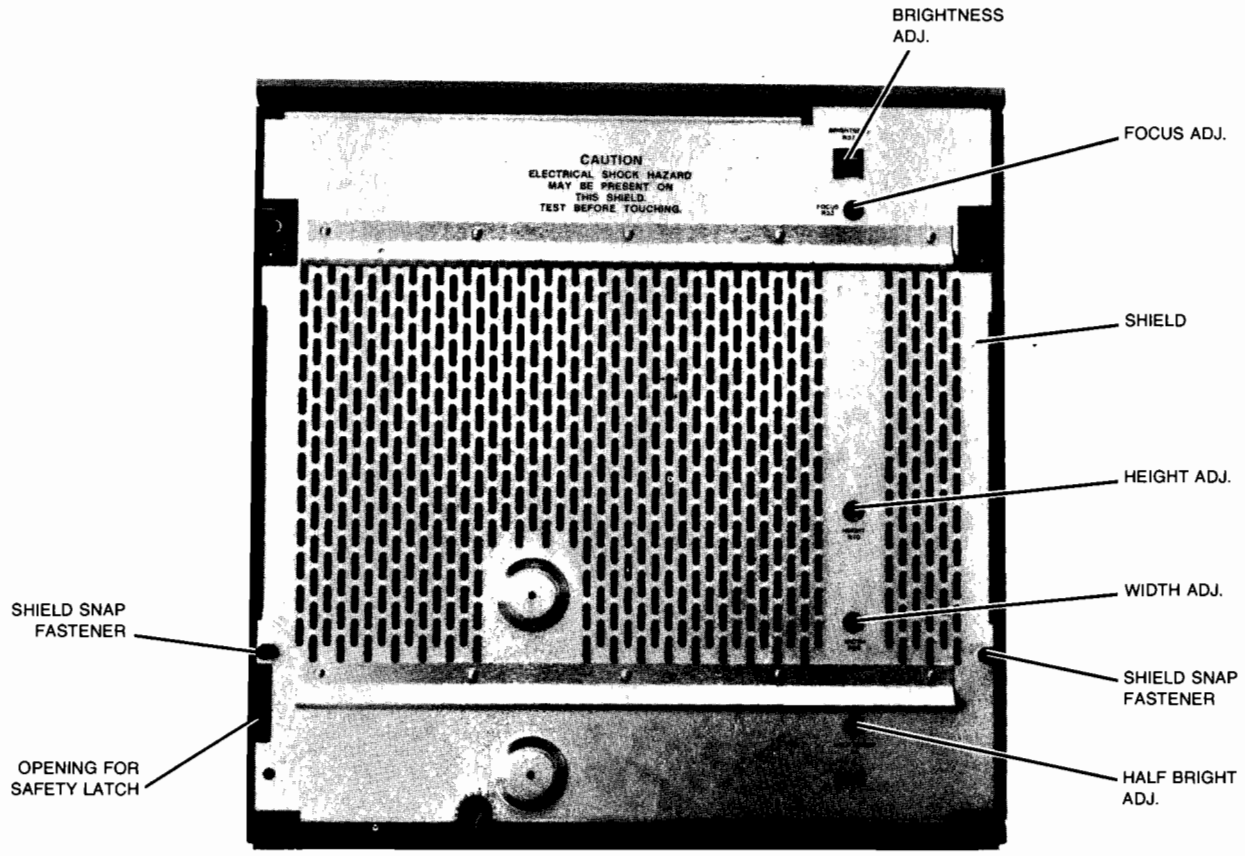


Figure 11-2. Mainframe Part Locations (Sheet 2 of 2)

GROUNDING REQUIREMENTS

To protect operating personnel, the National Electrical Manufacturer's Association (NEMA) recommends that the terminal's frame be grounded. The terminal is equipped with a three-conductor power cable which, when connected to an appropriate power receptacle, grounds the frame of the terminal. To preserve this protection feature, do not operate the terminal from an ac power outlet with no ground connection.

SELECTING LINE VOLTAGE AND FREQUENCY

The terminal can be operated from either 115 or 230V, 60 Hz line voltage (115 or 230V, 50 Hz optional). When shipped from the factory, the line voltage for which the terminal is configured is stamped on the mainframe rear panel identification label. If it is necessary to change the operating line voltage, ensure that power cord is disconnected and proceed as follows:

1. Open terminal to its half open position.
2. Remove power supply cover by removing the screw at the front of the cover and pulling the cover up and out of the mainframe.
3. Select the operating voltage by inserting the proper fuses into the appropriate locations shown in figure 11-3. For 115 volts, use a 0.5A, SB, 250V fuse and a 4A, SB, 250V fuse. For 230 volts, use a 0.20A, SB, 250V fuse and a 2A, SB, 250V fuse.
4. If changing from 60 Hz to 50 Hz operation or vice versa, set the HZ switch on the Display Timing/Control PCA to the appropriate position; closed for 50 Hz and open for 60 Hz.
5. Check, and if necessary, adjust power supply (refer to "Power Supply Adjustment").
6. Replace power supply cover, and secure in place with the screw.
7. Lower top cover to its closed position.
8. Perform terminal self-test (refer to Self Test section).

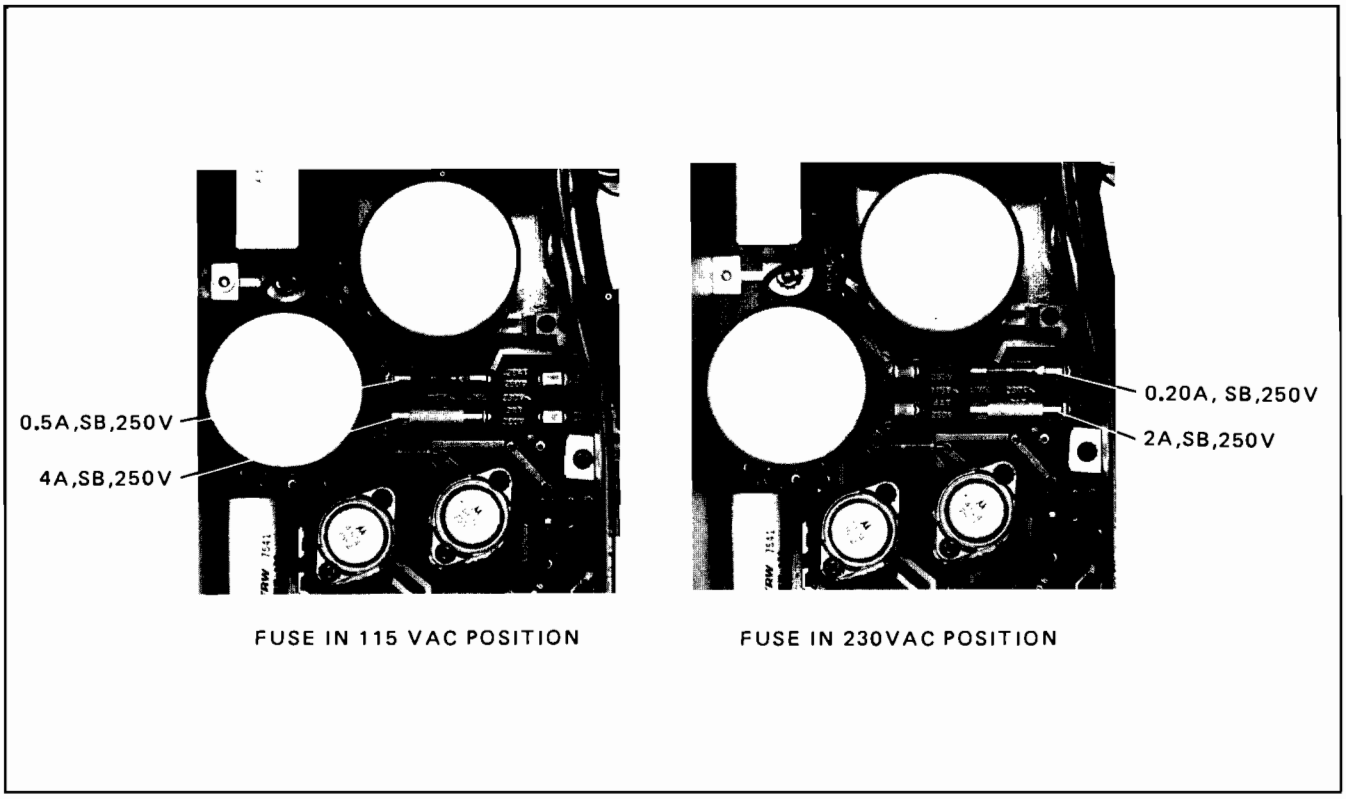


Figure 11-3. Fuse Positions for 115 AC and 230 VAC Line Voltage

POWER SUPPLY ADJUSTMENT

After installing or removing any device from the terminal, you should adjust the +5 volt output of the terminal's power supply. Only this voltage need be adjusted because the +5 volts provides reference for the other supply voltages. The adjustment requires a 20,000 ohms/volt voltmeter.

To adjust the +5V, proceed as follows:

1. Open the terminal and remove power supply cover.
2. Turn on ac power to terminal.
3. Check the voltages at the following points with the multimeter. (See figure 11-4.)

TEST POINT	VOLTAGE TOLERANCE
+ 5V diode	+4.85V to +5.25V
−42V diode	−40.0V to −46V
+12V diode	+11.8V to +12.6V
−12V diode	−11.8V to −12.6V

4. Adjust +5 volt potentiometer until all voltages are within tolerance.
5. When all voltages are within tolerance, turn off power, disconnect multimeter, and replace power supply cover.

EXTERNAL DEVICE INSTALLATION

The following paragraphs describe installation of devices external to the terminal including an external disc drive, installation of the diskette in the drive, the HP 13265A 300 Baud modem, and the HP 13349A Printer subsystem. After installation or removal of any device which uses power from the terminal power supply, the power supply output should be checked and adjusted, if necessary. Refer to the paragraph "Power Supply Adjustment" for instructions.

Installing a Disc Drive

You must connect the disc drive to the terminal with controller cable (13270-60004).

CAUTION

To ensure that your disc drive is operating at its optimum, locate the disc drive in a clean environment (free from dust and electromagnetic materials).

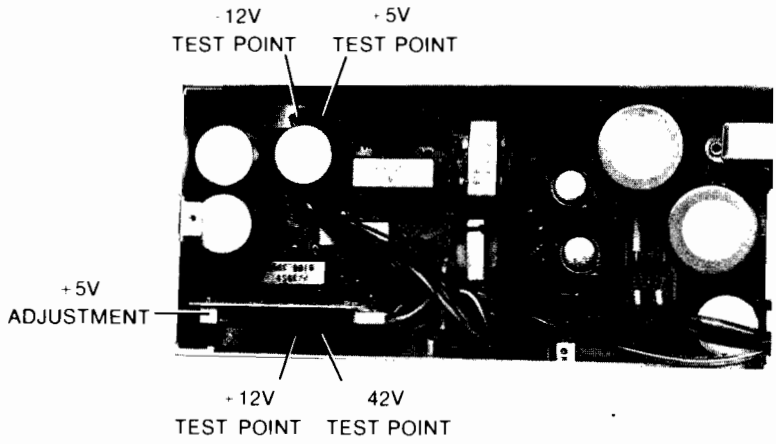


Figure 11-4. Power Supply Test Points

1. Set the terminal to the OFF position.

2. Place the disc drive unit next to the terminal on a flat surface such as a table, or desk, and remove the cable retainer as shown in figure 11-5.

Figure 11-5. Removing the Cable Retainer

3. Place one end of the controller cable onto the drive connector at the rear of the drive (figure 11-6). Note that both ends of the controller cable are identical, thus either end may be used.

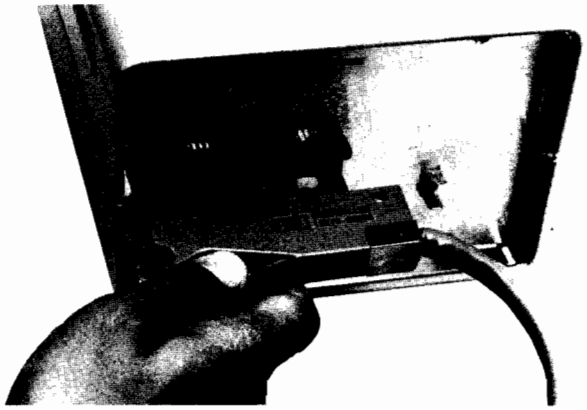


Figure 11-6. Connecting the Controller Cable to the Disc Drive

4. Secure the controller cable in place by inserting the cable retainer (figure 11-7) into its mating slots at the rear of the drive. Installing the cable retainer is a two step process: (1) insert the long-legged tabs into the two mating vertical slots on the left side (as viewed from the rear) and (2) insert the other tabs into the two mating horizontal slots.

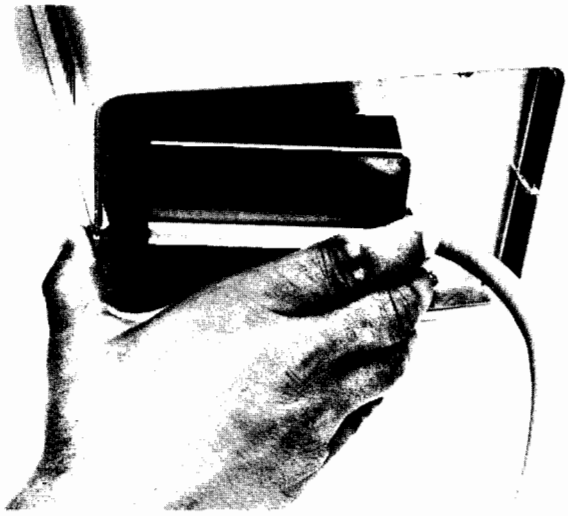


Figure 11-7. Installing the Cable Retainer

5. Connect the other end of the controller cable to the controller PCA in the terminal. Secure the connector to the PCA with the retainer and screw. Close the terminal door and secure it in place by tightening the two door latch screws.

6. Set the terminal power switch to the ON position.

7. Insert a diskette into the drive (refer to "Loading the Diskette").

8. Perform the disc self-test (refer to the Self Test section).

Loading the Diskette

CAUTION

When handling the diskette, care must be exercised not to touch the magnetic surface or to bend the diskette. Failure to do so may damage the diskette and can result in data loss and damage to the read/write heads of the disc drive.

To load the diskette into the disc drive, complete the following steps:

1. Access the disc loading slot by pushing upward on the door latch (figure 11-8).

CAUTION

Ensure that the door latch is fully opened to prevent damage to the read/write heads when inserting and removing the diskette.

2. Remove the diskette from its protective jacket and with the write-protect slot oriented to your left, insert the diskette into the disc loading slot until it reaches the backstop.

3. Close the door latch by pushing down until it clicks to complete the diskette loading.

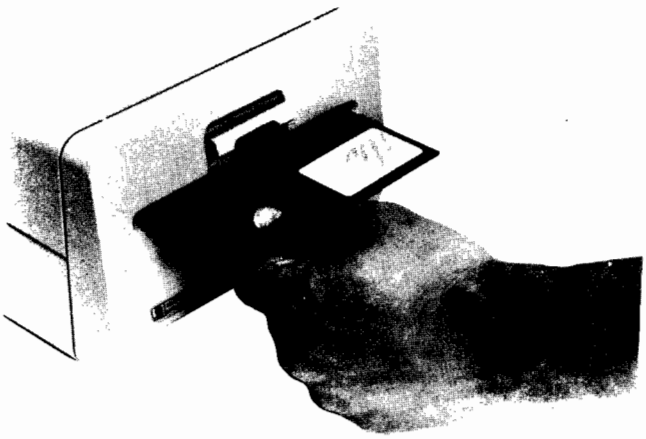


Figure 11-8. Inserting the Diskette into the Drive

HP 13265A 300 Baud Modem

The 13265A 300 Baud Modem is available, as an option, for use with the terminal. If it is used, the 13260A datacomm PCA (part no. 02640-60239) must be checked for proper configuration of the four switches located on it. Refer to "Terminal and Datacomm Configuration" in the Configuration section for configuration instructions. Details on how to install and use the modem are contained in the *HP 13265A 300 Baud Modem User Manual,* part no. 13265-90001.

INSTALLING AN HP-IB NETWORK

The following paragraphs describe cabling requirements and limitations, list the devices which can be connected into an HP-IB network, and supply procedures for installation and testing an HP-IB network.

Cabling Considerations

The total length of interconnecting cable in an HP-IB network is limited to an average of two meters per device up to a maximum of 20 meters. (Exceptions to this rule are described later in this discussion - refer to "Cabling Limitations".) Up to 15 devices are allowed in a network, with the terminal itself counted as a device.

Means are available for simulating up to seven additional devices in the network. This allows use of two meters of additional cabling for each simulated device. For example, a network containing only a terminal and one other device can use up to 18 meters of cable by including seven simulated devices in the network.

Devices are simulated using six IC-chip sockets and three IC chips on the HP-IB/ROM PCA. The six sockets are arranged in two rows of three each, one along the top edge of the PCA and one along the end edge (see figure 11-9). Each of the three sockets in each row accepts a 1-load, 2-loads, or 4-loads IC chip. (Each chip is labeled with the number of devices simulated.)

The number of devices simulated in the network is selected by inserting IC chips in the sockets of the top row such that, when the numbers associated with sockets containing an IC chip are added, the total equals the number of devices to be simulated.

For each empty IC socket in the top row, the socket labeled with the same number in the end row must be filled with an IC. The IC chips must be inserted on this PCA with the notched end of the chip on the right.

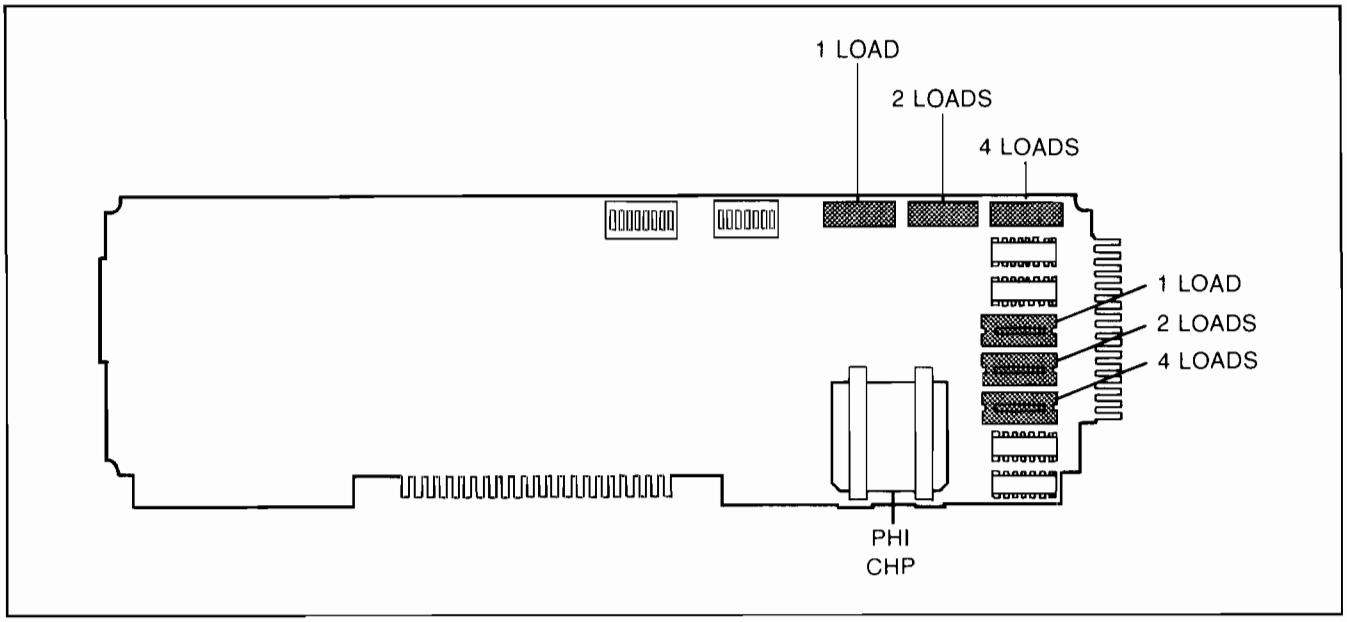


Figure 11-9. HP-IB/ROM PCA Load Sockets

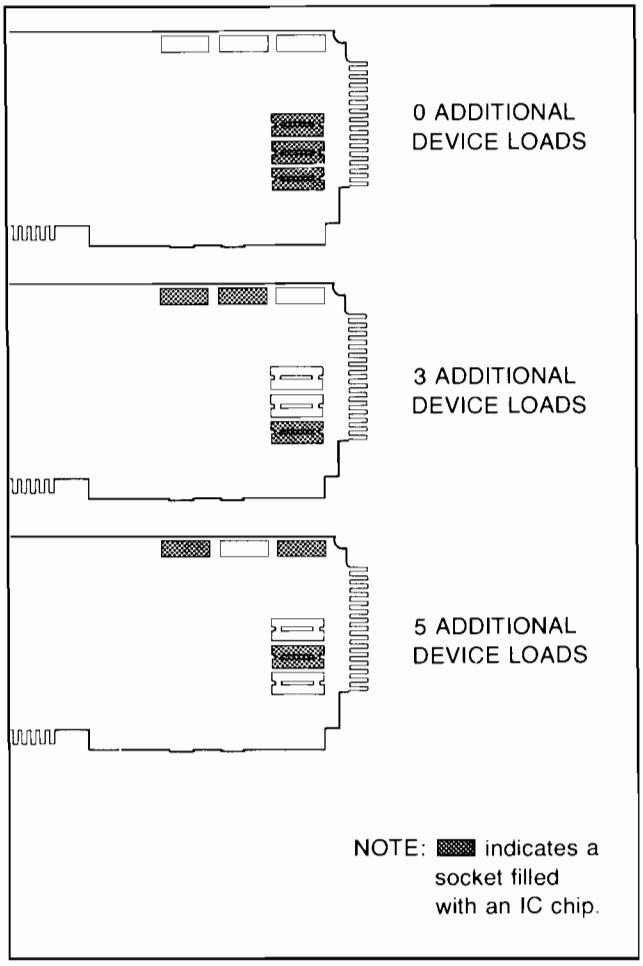


Figure 11-10. Simulating Devices in an HP-IB Network

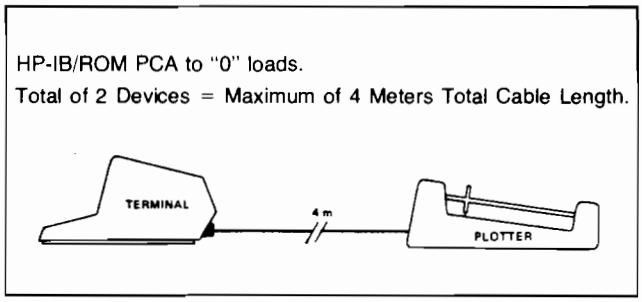
If more than one terminal is sued in the network, only one terminal (preferably the system controller) can have IC chips in its device-simulating IC-chip sockets. The sockets for all other terminals must be empty. This is true also for the IC sockets on the HP-IB Interface Adapter on any HP 2647A terminals in the network.

CAUTION

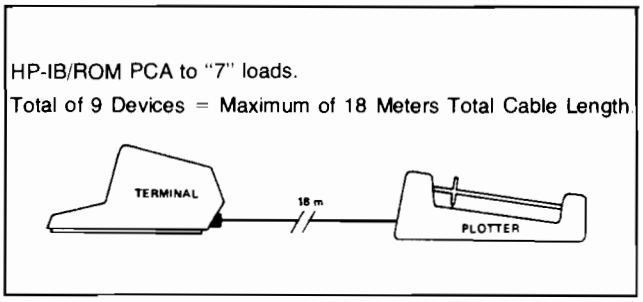
The three IC chips are not interchangeable. The IC chip for selecting one additional device load must always be in either the upper or lower "1 LOAD" socket, the IC chip for selecting two additional device loads must always be in either the upper or lower "2 LOADS" socket, and the IC chip for selecting four additional device loads must always be in either the upper or lower "4 LOADS" socket. The part number of the three IC chips are as follows:

1810-0408 (1 LOAD)
1810-0410 (2 LOADS)
1810-0409 (4 LOADS)

Figure 11-10 illustrates several possible connections. An IC chip extractor tool is included with the HP-IB Interface Adapter. For example, if you are connecting your terminal to an HP 2631A Printer and the HP-B/ROM PCA is set to "0", you would be limited to a total cable length of 4 meters as illustrated below:



By setting the HP-IB/ROM PCA to "7" instead, you could have up to 18 meters of cable between the two devices as illustrated below:



If you have a known combined length of HP-IB interconnecting cable and you want to know how many dummy device loads you must simulate in order to make use of all the available cable, use the following formula:

# of dummy loads = (total cable length/2) − # of actual devices

If this formula yields a fractional result, round the result up to the next higher integer value.

Cabling Limitations

Certain devices, when used in an HP-IB network, require that the maximum length of cabling in the network be limited to one meter per device (whether the device is actual or simulated) instead of the normal two meters per device. Such devices have a label near their HP-IB connector which calls attention to this limitation. In this case, the equation for determining the required number of devices to be simulated is as follows:

# of dummy loads = total cable length − # of actual devices

Devices Supported

The following peripheral devices are supported for connection in an HP-IB network:

- HP 9871A Printer
- HP 7310A Printer
- HP 2601A Printer
- HP 2671A Printer
- HP 9876A Printer
- HP 2673A Printer
- HP 2631A/B/G Printer
- HP 7225A Plotter
- HP 9872A/C/T Plotter
- HP 7245B Plotter/Printer
- HP 7225A/B Plotter
- HP 7580A Plotter
- HP 7470A Plotter
- HP 9111A Graphics Tablet
- HP 9874A Digitizer

Installation Procedures

Connecting an HP-IB network consists of configuring the switches on the HP-IB/ROM PCA as described in the Configuration section, selecting any simulated devices as described previously in the paragraph "Cabling Considerations", connecting the cables to the devices, and configuring any configuration switches on the devices.

Local HP-IB Testing

You can test the HP-IB Interface of any terminal in the configuration locally at any time by entering the following in the command channel of the particular terminal:

```
*TEST HP-IB↵
```

where * is the command channel prompt.

This test and its resultant messages are described fully in the Self-Test section of this manual.

Of particular interest at this point is that you can use this test to find out if your terminal is the System Controller and/or currently the CIC (the message displayed if this test is successful tells you this information as well as your terminal's HP-IB device address).

Remote HP-IB Testing

When a configuration includes multiple terminals, you can test the cable connections by performing terminal loop-back tests. With this test, data is sent from one terminal to another and the data is then sent back for comparison. You invoke this test by issuing the following command:

```
*TEST TERMINAL#n
```

where * is the command channel prompt and n is the address of the remote terminal with which the cable connection is to be tested.

This test and its resultant message are described fully in the Self-Test section of this manual.

NON-HP-IB CABLING

The following paragraphs discuss the HP 13232 family of cables and supply information on fabricating your own data communication cable.

HP 13232 Cable Assemblies

The HP 13232 cable assemblies provide interface connections between the terminal and modems, printers, and computers. Table 11-1 lists the particulars of each cable.

Table 11-1. 13232 Cable Assemblies

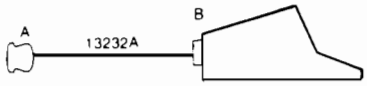
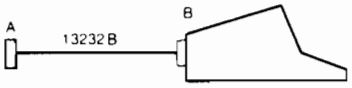
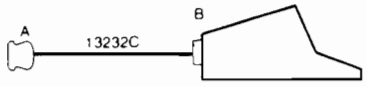
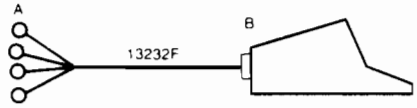
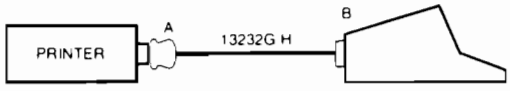
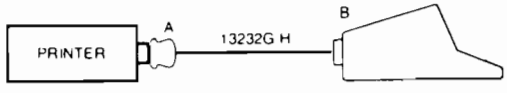
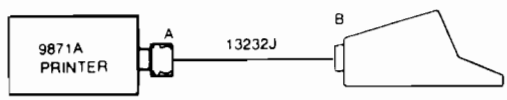
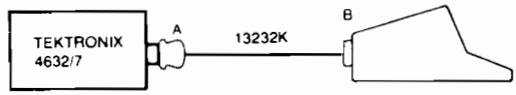
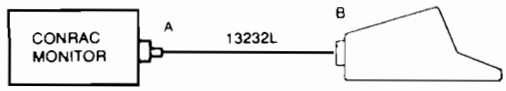
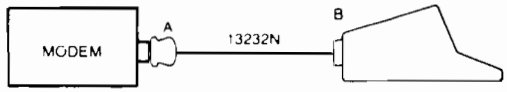
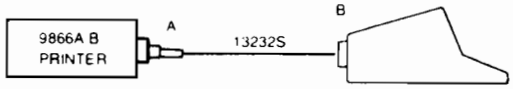
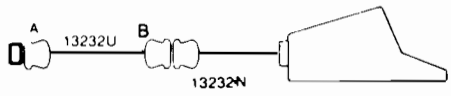
CABLE	FUNCTION	CONNECTORS A	CONNECTORS B	LENGTH	
13232A	Connects data communications interface PCA to modem 103/202. (Cable part no. 02640-60043.)	RS232 (male)	Hood	4.57 meters 15 feet	
13232B	Connects 12531/12880 teleprinter interface PCA to terminal. (Cable part no. 02640-60058.)	Hood	Hood	15.25 meters 50 feet	
13232C	Connects data communications interface PCA to RS232 connector. (Cable part no. 02640-60059.)	RS232C (female)	Hood	1.52 meters 5 feet	
13232F	Provides current loop connections for 13260B data communications interface. (Cable part no. 02640-60097.)	4 terminal lugs	Hood	1.52 meters 5 feet	
13232G	Connects 13250A Serial Printer Interface to RS232 compatible printers. (Cable part no. 02640-60098.)	RS232C (male)	Hood	4.57 meters 15 feet	
13232H	Same as 13232G. (Cable part no. 02640-60099.)	RS232C (female)	Hood	4.57 meters 15 feet	
13232J	Connects 13238A Duplex Register PCA to 9871A Printer. (Cable part no. 02640-60116.)	9871A printer (female)	Hood	1.83 meters 6 feet	

Table 11-1. 13232 Cable Assemblies (continued)

CABLE	FUNCTION	CONNECTORS A	CONNECTORS B	LENGTH	
13232K	Connects 13254A Video Interface PCA to Tektronix 4632/7 Video Copier. (Cable part no. 02640-60120.)	RS232 (male)	Hood	4.57 meters 15 feet	
13232L	Connects 13254A Video Interface PCA to Conrac Monitor. (Cable part no. 02640-60121.)	BNC	Hood	7.61 meters 2.5 feet	
13232N	Connects data communications interface PCA to modem. (Cable part no. 02640-60131.)	RS232C (male)	Hood	4.57 meters 15 feet	
13232S	Connects 13238A Duplex Register PCA to 9866A/B Printer. (Cable part no. 02640-60135.)	9866 printer (female)	Hood	1.83 meters 6 feet	
13232U	Provides direct connection to a computer by replacing the modem connections. (Cable part no. 5060-2403.)	RS232C (female)	RS232C (female)	1.52 meters 5 feet	

Fabricating Your Own Data Communications Cable

PCA hood connectors, RS232C connectors, and cables are available if you should need to fabricate your own data communications cable. The following text, figures, and tables provide pin-to-pin wiring information, names of signals, logic levels for the signals, and maximum cable lengths, hardware part numbers for the cable components, and fabrication instructions.

INTERFACE SIGNALS. Figures 11-11 through 11-13 show the cable connections at both the terminal and modem/computer ends of the cable and the signals on each conductor of the cables. Table 11-2 provides additional information on the signals and table 11-3 supplies signal voltage levels.

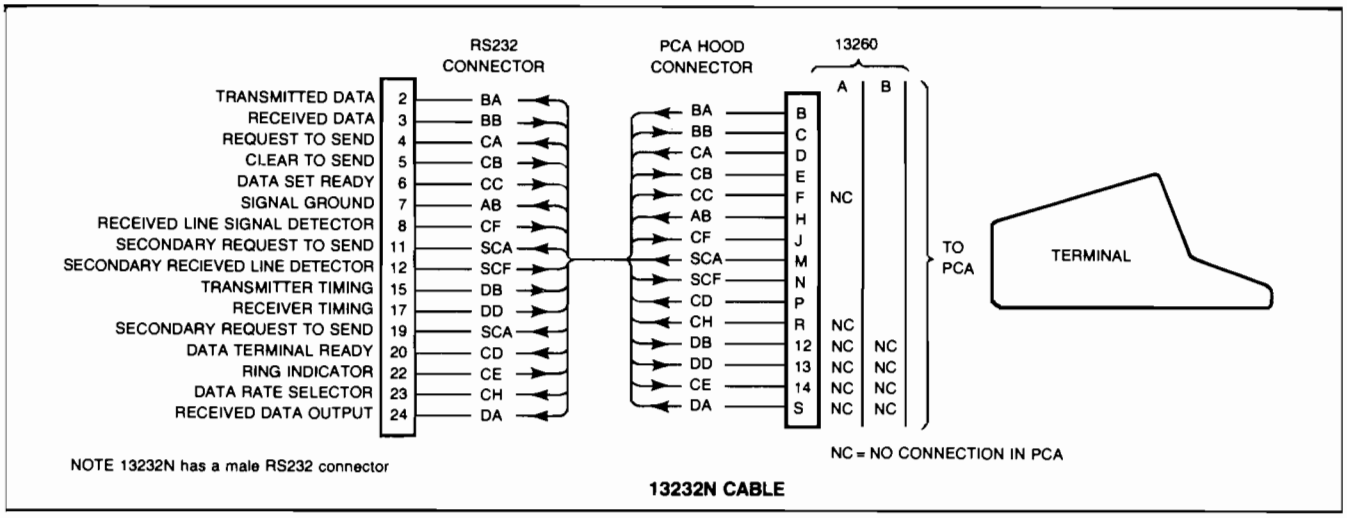


Figure 11-11. Point-to-Point Communications Cabling

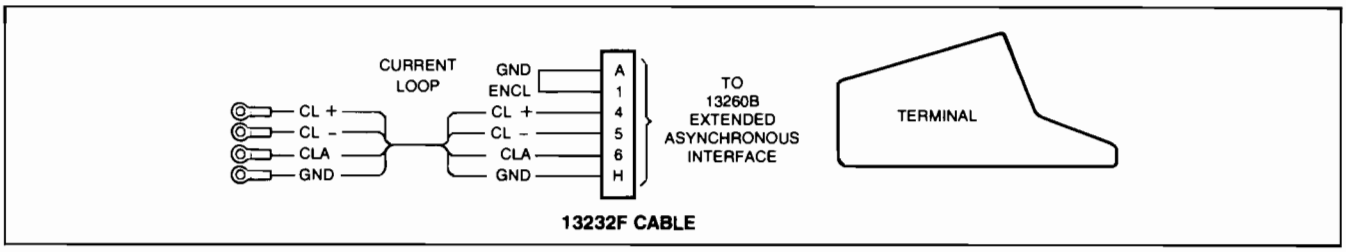


Figure 11-12. Current Loop Cabling

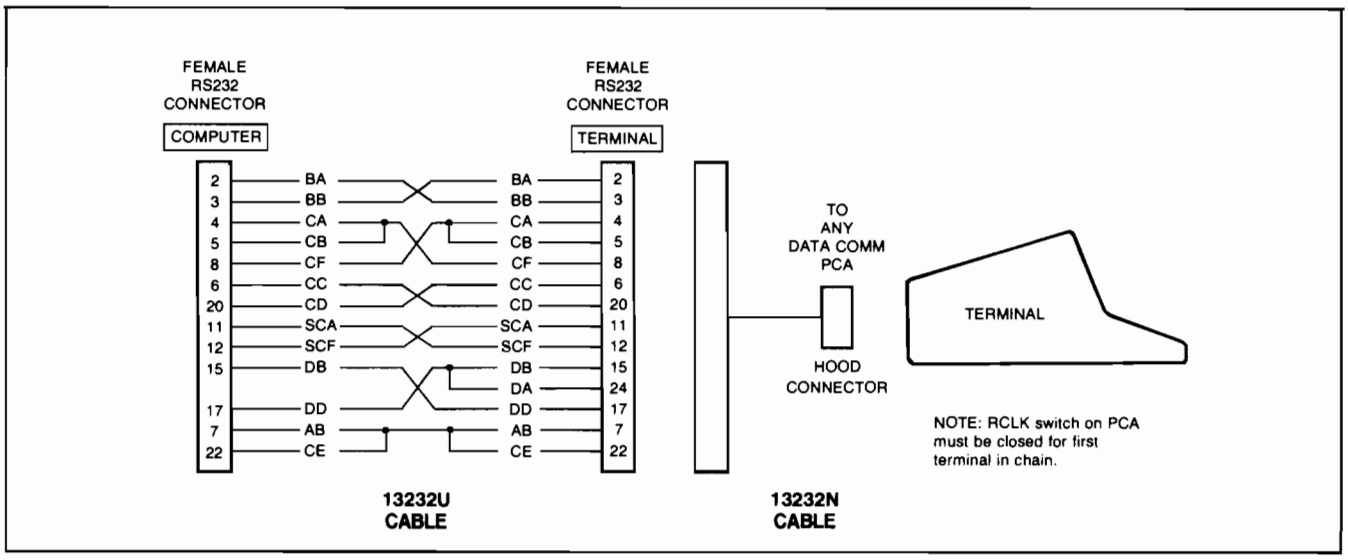


Figure 11-13. Modem By-Pass Cabling

Table 11-2. EIA RS232C and CCITT V24 Interface Data and Control Signals

CONNECTOR			CIRCUIT			MODEM			SIGNAL TYPE		
RS232	P2 13260 A	P2 13260 B	RS232	CCIV T2 T4	DESCRIPTION	TO	FROM	GND	DATA	CONTROL	TIMING
—	A	A	AA	—	Protective Ground			X			
7	H	H	AB	102	Signal Ground/Common Return			X			
2	B	B	BA	103	Transmitted Data	X			X		
3	C	C	BB	104	Received Data		X		X		
4	D	D	CA	105	Request to Send	X				X	
5	E	E	CB	106	Clear to Send		X			X	
6	F	F	CC	107	Data Set Ready		X			X	
20	P	P	CD	108.2	Data Terminal Ready	X				X	
22	—	—	CE	125	Ring Indicator		X			X	
8	J	J	CF	109	Received Line Signal Detector		X			X	
—	—	—	CG	110	Signal Quality Detector	X				X	
23	R	R	CH	111	Data Rate Selector (DTE Source)	X				X	
—	—	—	CI	112	Data Rate Selector (DCE Source)		X			X	
24	—	—	DA	113	Transmitter Timing (DTE Source)	X					X
15	—	—	DB	114	Transmitter Timing (DCE Source)		X				X
17	—	—	DD	115	Receiving Timing						X
—	—	—	SBB	118	Secondary Transmitted Data	X			X		
—	—	—	SBB	119	Secondary Received Data		X		X		
19	M	M	SCA	120	Secondary Request to Send	X				X	
—	—	—	SCB	121	Secondary Clear to Send		X			X	
12	N	N	SCF	122	Secondary Received Line Detector		X			X	

Table 11-3. Data Communications Signal Levels

DATA:		
Name	Space	Mark
Logic	0	1
Voltage	< +3V but > +25V	< −3V but > −25V
CONTROL:	ON (true)	OFF (false)
CLOCK SIGNALS:	0 = ground	1 = +5V

MAXIMUM CABLE LENGTHS. There are maximum length limitations on each type of cable. The following may be used as a guide for length considerations.

Modem/Computer to first terminal: 50 feet (RS232C standard)

Modem/Computer to terminal: 1000 feet (current loop on 13260B)

Terminal to terminal —
Note: Maximum total distance 16,000 feet

Terminals/Line \ Bits/Sec.	2400	4800	9600
4	2000 ft.	2000 ft.	2000 ft.
8	2000 ft.	2000 ft.	1200 ft.
16	1200 ft.	1200 ft.	480 ft.
32	1200 ft.	480 ft.	120 ft.

ASSEMBLY INSTRUCTIONS. Components needed to make a cable and their part numbers are listed in table 11-4. Figures 11-16 and 11-17 show exploded views of the PCA hood connector used on the terminal end of the cable and the RS232C connector used on the other end. Instructions for connecting the cable to the connectors are also contained in the figures.

Table 11-4. Parts for Fabricating Your Custom Data Communications Cable

ITEM	HP PART NO.	DESCRIPTION
RS232 Connector	5061-2405	See figure 11-14
PCA Hood Connector	5061-1340	See figure 11-15
PCA Hood to RS232 Connector Cable	8120-1903 or 8120-1930	26 AWG (or greater) Low Voltage Computer Cable

Note: All connectors include contacts.

PCA INSTALLATION

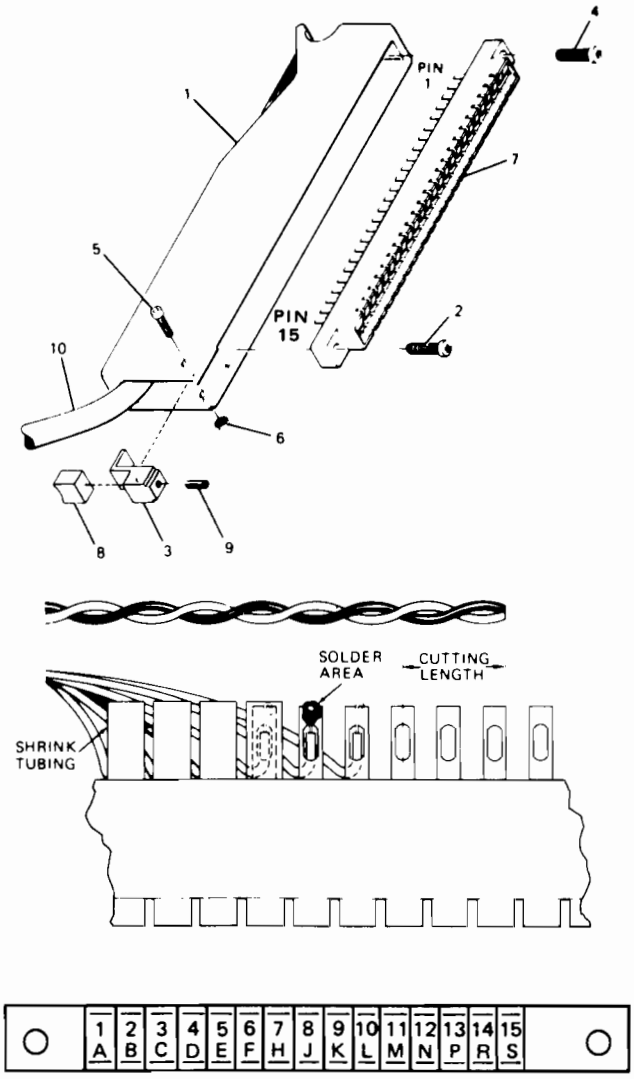
Before installing a PCA, the PCA should be configured as described in the Configuration section. After installation, the terminal self test should be performed (refer to the Self Test section) to ensure that the terminal operates properly. Also, the power supply output should be checked after a PCA is installed in the terminal (refer to "Power Supply Adjustment" earlier in this section for instructions).

HP 13238A Terminal Duplex Register

The HP 13238A Terminal Duplex Register accessory provides 8-bit parallel interface to external printers.

To install the HP 13238A accessory, perform all the following steps except steps 4 and 5.

1. Open terminal to its half open position (refer to "Opening the Terminal").
2. Configure the Terminal Duplex Register as described in the Configuration section.
3. Install Terminal Duplex Register PCA in first vacant Backplane Assembly connector adjacent to existing PCAs.

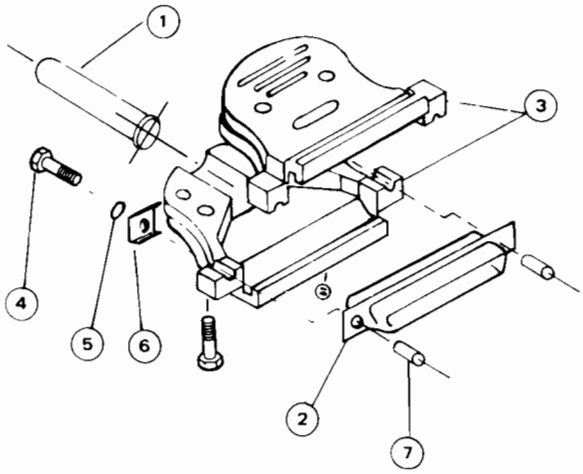


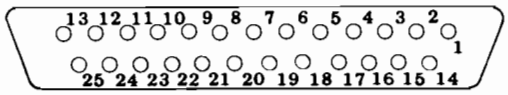
VIEWED FROM SOLDER SIDE

Assembly Procedures:

1. Insert approximately 10 inches of cable (item 10) into the connector hood (item 1).
2. Strip the outer jacket of the cable back 5 inches.
3. Remove approximately 1/4-inch of insulation from each signal wire.
4. Starting at the end of the 30-pin connector (item 7) nearest pins S and 15, solder the signal wires to the appropriate pins on the connector, and insulate each pin with tubing as shown at left.
5. Install the 30-pin connector in the connector hood using the two self-tapping screws (items 2 and 4).
6. Install the cable clamp (items 3 and 8), and tighten it in place with the screw and nut (items 5 and 6).
7. Tighten the cable clamp on the cable with the setscrew (item 9).

Figure 11-14. Assembling the PCA Hood Connector





VIEWED FROM SOLDER SIDE

Assembly Procedure:

1. Slide rubber bushing (item 1) over end of cable, leaving about 6 inches of cable end exposed for wire stripping, etc.
2. Strip back the cable jacket 1-inch.
3. Clip the unused conductor wires to the edge of the cable jacket.
4. Remove 1/4-inch of insulation from the ends of the conductor wires to be used.
5. Solder the conductor wires onto the contacts of the contact assembly (item 2). (Select either the male or female contact assembly provided for your particular application.)
6. Slide the rubber bushing to the end of the cable such that the rubber bushing flange is flush with the stripped end of the cable jacket.
7. Assemble the two halves of the connector (item 3) onto the contact assembly (item 2). (Use the screws and nuts provided.)
8. Mount the two screws, threaded spacers, and other hardware (items 4 thru 7) onto the contact assembly.

Figure 11-15. Assembling the RS232C Connector

4. Open mainframe rear door by twisting two lock extrusions.

5. Holding the Terminal Duplex Register PCA in place, connect the hood connector of the cable assembly, supplied with the printer subsystem, to PCA connector P2.

NOTE: The hood connector and PCA connector P2 and identically keyed to prevent inadvertent erroneous connections. Connecting the two together requires minimal hand pressure. If excessive resistance is encountered, an incorrect connection is being attempted.

For printer interfacing information refer to the *HP 9866A/B Printer Operator's Manual,* part no. 09866-90901, or the *HP 13349A Printer Subsystem Operating Manual,* part no. 13349-90901.

6. Check, and if necessary, adjust power supply (refer to "Power Supply Adjustment").

7. Close the terminal.

HP 13250B Serial Printer Interface

The HP 13250B provides an RS232C interface to serial printers for alphanumeric output. You can configure the 13250B to be compatible with many RS232 serial printers requiring handshake or full-character protocol. For details on configuring and installing the interface, refer to the *HP 13250 Accessory Manual,* part no. 02640-90042.

HP 13254A Video Interface

The 13254A consists of a Video Interface PCA, part no. 02640-60019, and a Sweep Extender Cable, part no. 02640-60122. This accessory is used to link the terminal to a compatible video monitor or hard copy device. The accessory requires one option slot. Detailed installation and operating information is contained in the *HP 13254A Accessory Manual,* part no. 13254-90001.

HP 13260B Extended Asynchronous Communications Interface

The terminal is configured at the factory with the standard asynchronous data communications PCA, part no. 02640-60239, already installed. Only one data communications PCA may be installed in the terminal at any time. To install the optional data communications PCA, part no. 02640-60143, proceed as follows:

CAUTION

MOS integrated circuits can be damaged by electrostatic discharge. Use the following precautions:

DO NOT wear clothing subject to static charge buildup, such as wool or synthetic materials.

DO NOT remove the circuit from its conductive foam pad until you are ready to install it.

AVOID touching the circuit leads. Handle by the plastic package only.

ENSURE that the circuit, work surface (table, desk, etc.) and PCA are all at the same ground potential. This can be done by touching the foam pad to the PCA and then touching the foam pad, circuit, and PCA to the work surface.

1. Perform complete terminal Self Test (refer to the Self-Test section) to verify proper terminal operation before installing the accessory.

2. Turn off LINE switch at rear of terminal and disconnect power cord.

3. Open the terminal.

4. Remove existing data communications PCA.

5. Configure new data communications PCA for your particular application by setting the switches on the PCA. (Refer to the Configuration section for instructions.)

6. Install new data communications PCA into vacated backplane assembly connector adjacent to existing PCAs. Reconnect ground cable assembly between power supply chassis ground and PCA ground connector lug.

NOTE: To ensure proper terminal operation, all PCAs must be installed in adjacent Backplane Assembly connectors. There should never be vacant connectors between PCAs.

7. Open mainframe rear door by twisting the two lock extrusions.

8. Connect Test Connector Assembly, part no. 02645-60002 (supplied with the terminal) to PCA, and perform the Data Comm Self-Test to verify proper operation of the PCA (refer to "Data Communications Self-Test" in the Self Test section). After self test is performed, remove the Test Connector Assembly.

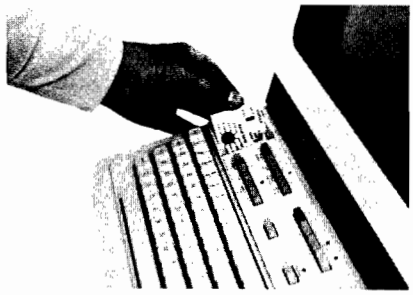
9. Connect datacomm cable to PCA.

NOTE: The hood connector and PCA connector P2 are identically keyed to prevent inadvertent erroneous connections. Connecting the two together requires minimal hand pressure. If excessive resistance is encountered, an incorrect connection is being attempted.

10. Install baudrate switch overlay and keyboard overlay as shown in figure 11-16.

11. Close the terminal.

A. REMOVING A KEYBOARD OVERLAY



B. INSTALLING BAUDRATE OVERLAY

C. INSTALLING KEYBOARD OVERLAY

Figure 11-16. Installing Keyboard Overlays

CONFIGURATION

SECTION 12

This section contains instructions for configuring all PCAs which might normally be used with the terminal. This includes those supplied with the standard terminal and those supplied as a part of an accessory package. The PCAs covered in this section are as follows:

Processor 02640-60249
Selects operating characteristics for the terminal and some datacomm operating characteristics.

Display Memory/DMA 02640-60250
Selects display options.

Display Timing/Control 02640-60267
Selects display options.

HP-IB/ROM 02640-60248
Selects operating characteristics for the HP-IB network.

13238A
Terminal Duplex Register accessory - used as an interface for any device using 8-bit parallel data transfer.

13250B
Serial Printer Interface/Extended Asynchronous Communications Accessory - used either with a serial printer or as an asynchronous communications interface.

13254A
Video Interface - connects terminal to either a video monitor or a video hardcopy device.

13260A
Standard asynchronous datacomm PCA.

13260B
Same as 13250B (same PCA).

TERMINAL AND DATACOMM CONFIGURATION

The switches which control the terminal and datacomm operating characteristics are located on the keyboard, the datacomm PCA, and the Processor PCA. Table 12-1 lists the functions of the three switches on the keyboard. If the 13265A 300 Baud modem is to be used with the terminal, the 02640-60239 datacomm PCA must be configured as shown in figure 12-1. The functions of the Processor PCA switches are listed briefly in table 12-2 and in detail in table 12-3. The functions of the switches on the 13260B datacomm PCA (part no. 02640-60143) are listed in table 12-4. Figures 12-2 and 12-3 locate the switches on the PCAs and figure 12-4 provides the configuration procedures for both the datacomm and Processor PCAs.

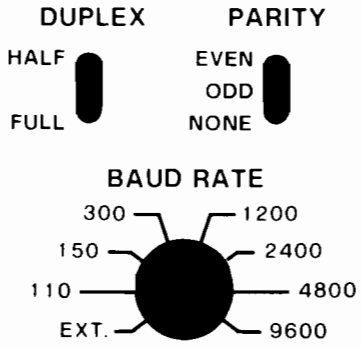


Table 12-1. Keyboard Communications Switches

DUPLEX Switch. HALF: Typed characters are processed by the terminal and transmitted to the computer. FULL: Typed characters are transmitted to the computer and not processed by the terminal until returned from the computer. (This function is ignored in Block Mode.)

PARITY Switch. When set to EVEN/ODD/NONE, even/odd/no parity is transmitted for each character. Incorrect parity: a "■" is displayed.

BAUD RATE Switch. Selects data transmission rate of 110, 150, 300, 1200, 2400, 4800, or 9600 baud. EXT: any rate between 110 and 9600 can be selected from an external source. The 110 baud rate uses two stop bits per character; all others use one stop bit.

REMOTE — When down, the terminal is in Remote (on-line) operation. Otherwise, the terminal is in local (off-line) operation.

BLOCK MODE — When the terminal is in Block Mode, typed data is displayed but not transmitted to the computer until requested by the computer or until after the ENTER key has been pressed and the computer has responded. Otherwise, the terminal is in Character Mode and data is transmitted as typed.

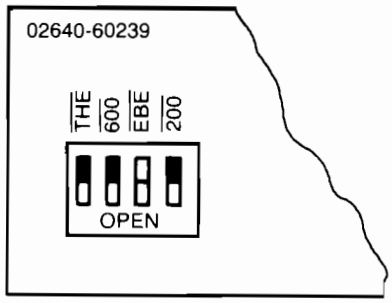
BREAK — In basic communications, transmits a BREAK signal to interrupt computer operation. (Transmits a 200 ms space on the asynchronous data communication line and sets secondary channel low for 200 ms.)

● TRANSMIT — The indicator will be lighted when a data link exists for transmission between the terminal and the computer.

ENTER

- Character Mode, Format Off. The entire line containing the cursor is transmitted as a block.
- Character Mode, Format On. Unprotected characters from the cursor position to the end of the unprotected field are block transmitted. The cursor is left at the first character position after the end of the field.
- Block Mode, Format Off. After receiving a DC1 from the computer, the terminal informs the computer by transmitting a DC2 control character (or DC2 with Line Strapping – see "Strapping Options") that the terminal is ready to transmit characters from the cursor to the end of the line of memory (dependent on Line or Page strapping).[1]
- Block Mode, Format On. After receiving a DC1 from the computer, informs the computer by transmitting a DC2 (or DC2 with Line Strapping) that the terminal is ready to transmit the current field, or all unprotected fields from the cursor to the end of memory, each delimited by a unit separator, (dependent on Line/Page strapping).[1]

[1]Basic Data Communications only



= CLOSED

SWITCH	SETTING	DESCRIPTION
THE	Closed	Disables Transmit Handshake
600	Closed	Disables 600 Baud Rate
EBE	Open	Enables External Baud Rate
200	Closed	Disables 200 Baud Rate

Note: Switches 200, 600, and EBE work only when the baud rate selector is set to EXT on the terminal. Only one of these three switches may be enabled (open) at any one time.

Figure 12-1. Standard Asynchronous Data Communications PCA Strapping for use with the HP 13265A 300 Baud Modem

Table 12-2. Processor PCA Switch Summary

SWITCH	CHARACTER PROTOCOL	BLOCK PROTOCOL
A	Function key transmission	(not used)
B	Space overwrite latch	same
C	Cursor end-of-line wraparound	same
D	Line/Page mode	same
E	Paper tape mode	same
F	Fast binary read	(not used)
G	Block transfer handshake	(not used)
H	Inhibit DC2	(not used)
J	Auto terminate	same
K	Clear terminator	same
L	Self-test inhibit	same
M	Reverse action of CNTL key with INSERT CHAR and DELETE CHAR keys (wrap function)	same
N	Escape code transfer to printer	same
P	(Not used)	same
Q	(Not used)	same
R	Circuit Assurance	Internal Data Set Ready
S	Main/Reverse Channel configuration. Switches S and T cannot be modified programmatically.	Space Compression
T		Output block size. (Switches T and U cannot be modified programmatically.)
U	CPU break	
V	Carrier detect	Synch Mode for Asynchronous Operation
W	Data Comm Handshake	same
X	Data speed select	same
Y	Transmit LED	same
Z	Force Parity	Transparency

Table 12-3. Processor PCA Strapping Options

STRAP	STRAPPING OPTION	NORMAL OPERATION (SWITCH CLOSED)	OPERATION WITH STRAPPING OPTION (SWITCH OPEN)
A	Function Key Transmission	The escape code sequence generated by the major function keys (such as, ROLL UP, ROLL DOWN, etc.) are executed locally, but not transmitted to the computer.	The escape code sequences generated by all keys are transmitted to the computer. If operating in half duplex, the function is also executed locally.
B	Space Overwrite (SPOW) Latch Enable	Spaces typed will overwrite existing characters.	When the SPOW latch is off, overwriting occurs as normal. When the SPOW latch is on, spaces cause the cursor to move forward but not overwrite any existing characters. The SPOW latch is turned on by a Carriage Return, and off by a Line Feed, Home or Tab.
C	Cursor End-of-Line Wraparound	At the end of each line, a local Carriage Return and Line Feed are generated; the cursor moves to the beginning of the next line.	A Carriage Return and Line Feed are not generated at the end of each line. The cursor remains in and overwrites column 80.
D	Line/Page	The terminal is set to transfer a line at a time in Block Mode.	Entire pages of information are transferred in Block Mode.
E	Paper Tape Mode	When the READ key is pressed with AUTO LF key latched down, each tape record begins with an ␍ and is terminated by by a ␚.	Each tape record is terminated by ␚␍.
F	Fast Binary Read	The transmission rate is determined by the BAUD RATE switch on the keyboard.	When an ␛ e (Fast Binary Read) is issued by the computer, the baud rate is switched automatically to 9600 baud (if the terminal is equipped with disc drives or cartridge tape units).
G	Block Transfer Handshake	In Block Mode, all data transfers to the computer are sent upon receipt of a DC1 from the computer.	All Block Mode transfers (i.e., cursor sense, terminal and device status, device I/O responses, display memory, and function keys) are preceded by a DC2. The terminal sends the DC2 upon receipt of a DC1 from the computer. After the CPU receives the DC2 from the terminal, another DC1 is required to trigger transmission of data from the terminal.
H	Inhibit DC2	During Block Mode Handshake transfers, the terminal sends a DC2 in response to a DC1 prior to sending data. (See Block Transfer Handshake strapping above.)	A DC1 from the computer is not required to trigger data transfers to the computer. Also, the DC2 from the terminal is not sent during Block Mode Transfer handshakes. (See Block Transfer Handshake strapping above.) Additionally, when the ENTER key is pressed in Block Mode the cursor will be placed in the first column before transmission occurs if operating in Line/Field Mode (switch D closed) or Home'd if operating in Page Mode (switch D open.) Opening both switches G and H eliminate the terminal's use of the Handshake protocol entirely.
J	Auto Terminate	No effect.	When in BLOCK mode and the ENTER key is pressed, places a non-displaying terminator before the cursor position.
K	Clear Terminator	No effect	Clear terminator caused by Strapping Option J or ␛—.
L	Self Test Inhibit	No effect.	Self Test function is inhibited. Pressing TEST key or issuing ␛ z displays the NO TEST message. TAPE TEST and SELF TEST functions are not affected. DATA COMM SELF-TEST is disabled.
M	INSERT and DELETE CHAR with wrap (Reverse Sense)	No effect.	Reverses effect of CNTL key on INSERT CHAR and DELETE CHAR keys (i.e., when key is pressed, line wrap around is in effect without having to press CNTL key. When either key is pressed while pressing CNTL, normal insert character and delete character functions are in effect.)

Table 12-3. Processor PCA Strapping Options (Continued)

STRAP	STRAPPING OPTION	NORMAL OPERATION (SWITCH CLOSED)	OPERATION WITH STRAPPING OPTION (SWITCH OPEN)
N	Escape Code Transfer to Printer	No effect.	Escape codes relating to the display (e.g., display enhancements, alternate character sets, format mode, fields, etc.) are sent to printer if it is selected as a destination device.
P,Q	(Not Used)		
R	Circuit Assurance	The transition from receive state to transmit state occurs after both CB (106) (Clear to Send) and SB (122) (Secondary Receive Data) go on within 2.6 seconds. Otherwise, the terminal returns to the receive state.	The transition from receive state to transmit state occurs after CB (106) (Clear to Send) goes on.
S,T	Main Channel Protocol	Reverse Channel protocol (both switches closed).	**S-closed, T-open:** Main channel with STX/ETX as Start of Data and End of Data. **S-open, T-closed:** Main channel with EOT as End of Data. **S-open, T-open:** Main channel with ETX as End of Data.
U	CPU Break	The CPU can interrupt the terminal while it is in the transmit state. The CPU initiates an ON to OFF transition of the SB(122) (Secondary Receive Data) line. The terminal responds by turning off CA (106) (Request to Send) and going to the receive state.	The terminal ingores all transitions on the SB (122) (Secondary Receive Data) line from the modem in the transmit state.
V	Carrier Detect	When the terminal is in the receive state, an ON to OFF transition of CF (109) (Carrier Detect) line from the modem causes the terminal to go into the transmit state. Transitions of CF have no effect while the terminal is in the transmit state.	Transitions of CF (109) (Carrier Detect) line have no effect on the terminal.
W	Data Comm Handshake	Enables ENG/ACK Handshake.	Enables XON/OFF (DC1/DC3) Handshake.
X	Data Speed Select	Holds data speed signal low (CH (111) = 0).	Sets data speed signal high (CH (111) = 1).
Y	Transmit LED	The TRANSMIT light on the keyboard is turned on when CB (106) (Clear to Send) line from the modem is high. It is turned off when the CB (106) line goes low.	The TRANSMIT light on the keyboard is turned on when the CC (107) (Data Set Ready) line from the modem is high and the 13260B Extended Asynchronous Communications Interface PCA is used. It is turned off when the CC line goes low.
Z	Parity	The PARITY switch on the terminal keyboard is affected as follows:	
		No Parity: Send 8 bits and receive 8 bits. Force bit 8 to zero. Check for parity error. **Odd Parity:** Send 7 data bits + odd parity. Receive 7 data bits + odd parity. Check for parity error. **Even Parity:** Send 7 data bits + even parity. Receive 7 data bits + even parity. Check for parity error.	**No Parity:** Send 8 bits and receive 8 bits. Force bit 8 to one on send. No check for parity error. **Odd Parity:** Send 7 bits + odd parity. Receive 7 bits. No check for parity error. **Even Parity:** Send 7 data bits + even parity. Receive 7 data bits. No check for parity error.

Table 12-4. Extended Asynchronous Communications Interface Strapping Options

<table>
<tr><th>STRAP</th><th>STRAPPING OPTION</th><th>DESCRIPTION</th></tr>
<tr><td>FC0
thru
FC7</td><td>(Not Used)</td><td>(This switch should always be open.)</td></tr>
<tr><td>B0
thru
B7</td><td>Custom Baud Rate Select</td><td>The switches are set to the binary equivalent of a number determined by the formula:
$INT(\frac{153600}{baud\ rate}) - 1$
(See example in figure 5-10.)</td></tr>
<tr><td>134</td><td>134.5 Baud</td><td>(This switch should always be open.)</td></tr>
<tr><td>S0
thru
S2</td><td>Transmit Baud Rate</td><td>
<table>
<tr><th colspan="3">SWITCH SETTING</th><th>TRANSMIT BAUD RATE</th></tr>
<tr><th>S0</th><th>S1</th><th>S2</th><th></th></tr>
<tr><td>O</td><td>O</td><td>O</td><td>Transmit baud rate = receive baud rate.</td></tr>
<tr><td>O</td><td>C</td><td>C</td><td>110</td></tr>
<tr><td>C</td><td>O</td><td>C</td><td>150</td></tr>
<tr><td>O</td><td>O</td><td>C</td><td>300</td></tr>
<tr><td>C</td><td>C</td><td>O</td><td>1200</td></tr>
<tr><td>O</td><td>C</td><td>O</td><td>2400</td></tr>
<tr><td>C</td><td>C</td><td>C</td><td>Custom</td></tr>
<tr><td colspan="4">O = open, C = closed</td></tr>
</table>
</td></tr>
<tr><td>CBE</td><td>Custom Baud</td><td>Closed: Enables custom receive baud rates. (The keyboard BAUD RATE switch must be set to EXT.)
Open: Receive baud rate is set by keyboard BAUD RATE switch.</td></tr>
<tr><td>B8
thru
B11</td><td>Custom Baud Rate Select</td><td>The switches are set to the binary equivalent of a number determined by the formula:
$INT(\frac{153600}{baud\ rate}) - 1$</td></tr>
<tr><td>A4,A9
thru
A11</td><td>Module Address</td><td>Provides PCA address so that firmware can address the PCA. These switches should always be set to 10, (A4 open, A9 thru A11 closed).</td></tr>
<tr><td>IAT</td><td>Inhibit Attention</td><td>(This switch must be closed when receive handshake is used.)</td></tr>
<tr><td>ATN2</td><td>Enable Attention Two</td><td>(This switch should always be open.)</td></tr>
<tr><td>THE</td><td>Transmit Handshake Enable</td><td>Closed: Permits the associated external device (a or computer) to signal a "busy" condition on CB (Clear to Send) or SCF (Secondary Carrier) control lines and temporarily stop data transmission from the terminal.
Open: Transmit Handshake disabled.</td></tr>
<tr><td>RHE</td><td>Receive Handshake Enable</td><td>Closed: Permits the terminal to signal a "busy" condition on the CD (Data Terminal Ready) control line and temporarily stop data transmission from the associated external device (a computer).
Open: Receive Handshake Disabled.</td></tr>
<tr><td>NOSB</td><td>SCF Inhibit</td><td>Closed: Inhibits RS232 SCF (Secondary Carrier) control line.
Open: Enables RS232 SCF (Secondary Carrier) control lines.</td></tr>
<tr><td>2SB</td><td>Stop Bit Select</td><td>Selects the number of stop bits to be appended to the data bits during transmission.
Closed: Selects 2 stop bits.
Open: Selects 1 stop bit.
NOTE: Selecting 110 baud automatically appends 2 stop bits.</td></tr>
</table>

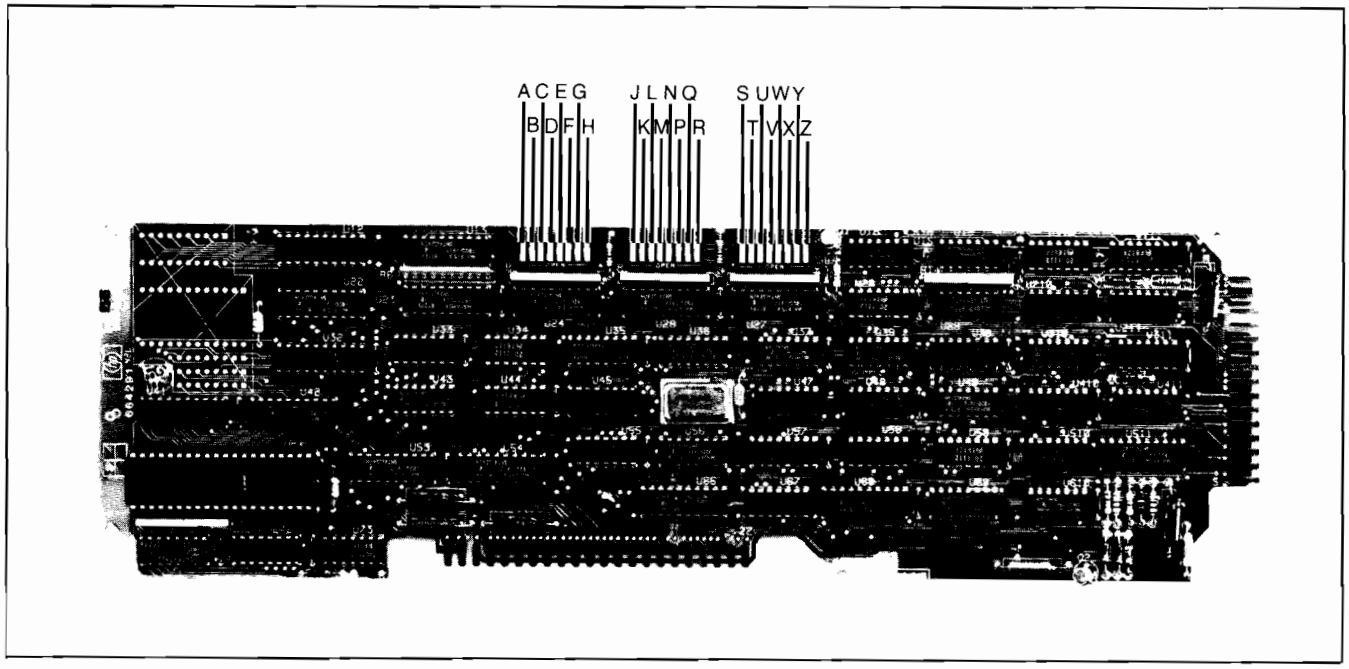


Figure 12-2. Processor PCA Configuration Switches

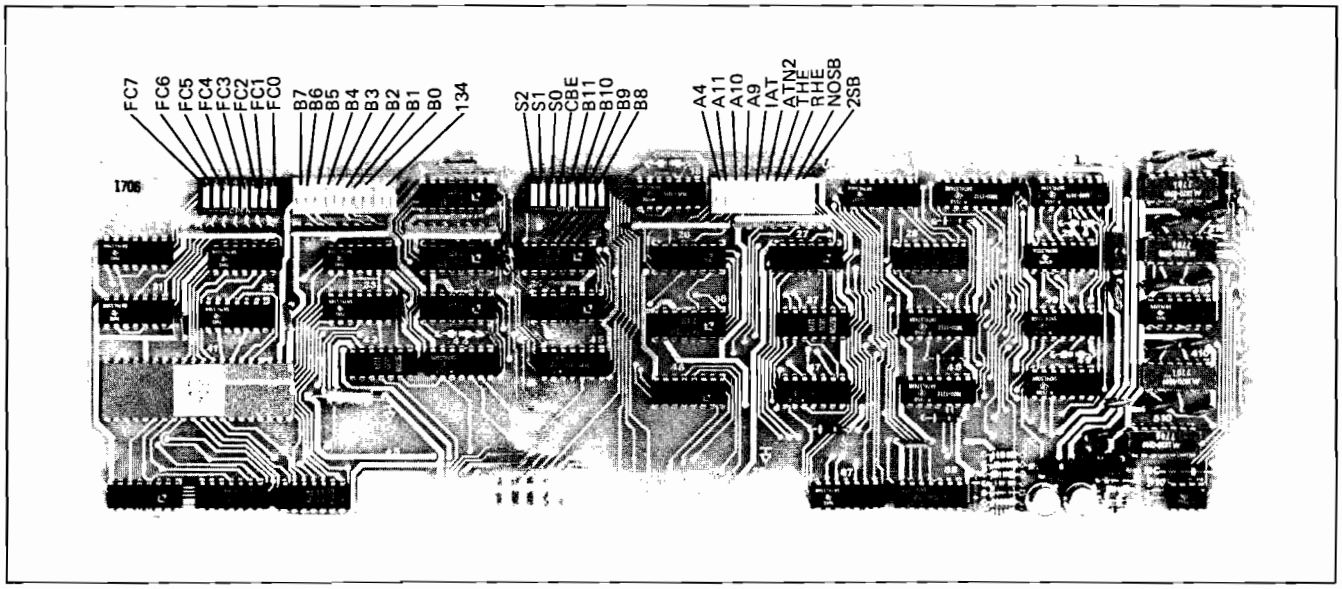


Figure 12-3. Extended Asynchronous Communications PCA Configuration Switches

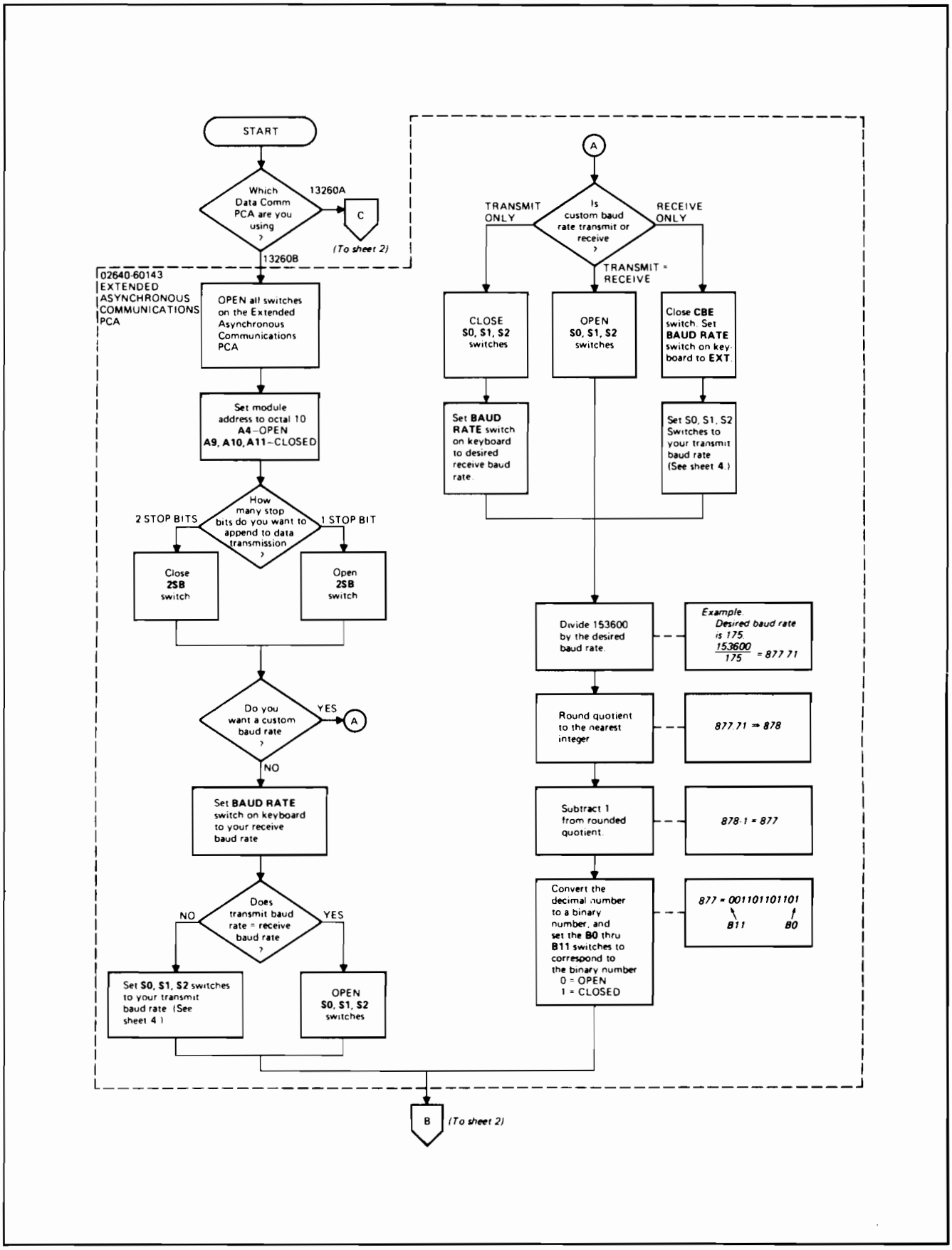


Figure 12-4. Data Communications Configuration Flowchart (Sheet 1 of 4)

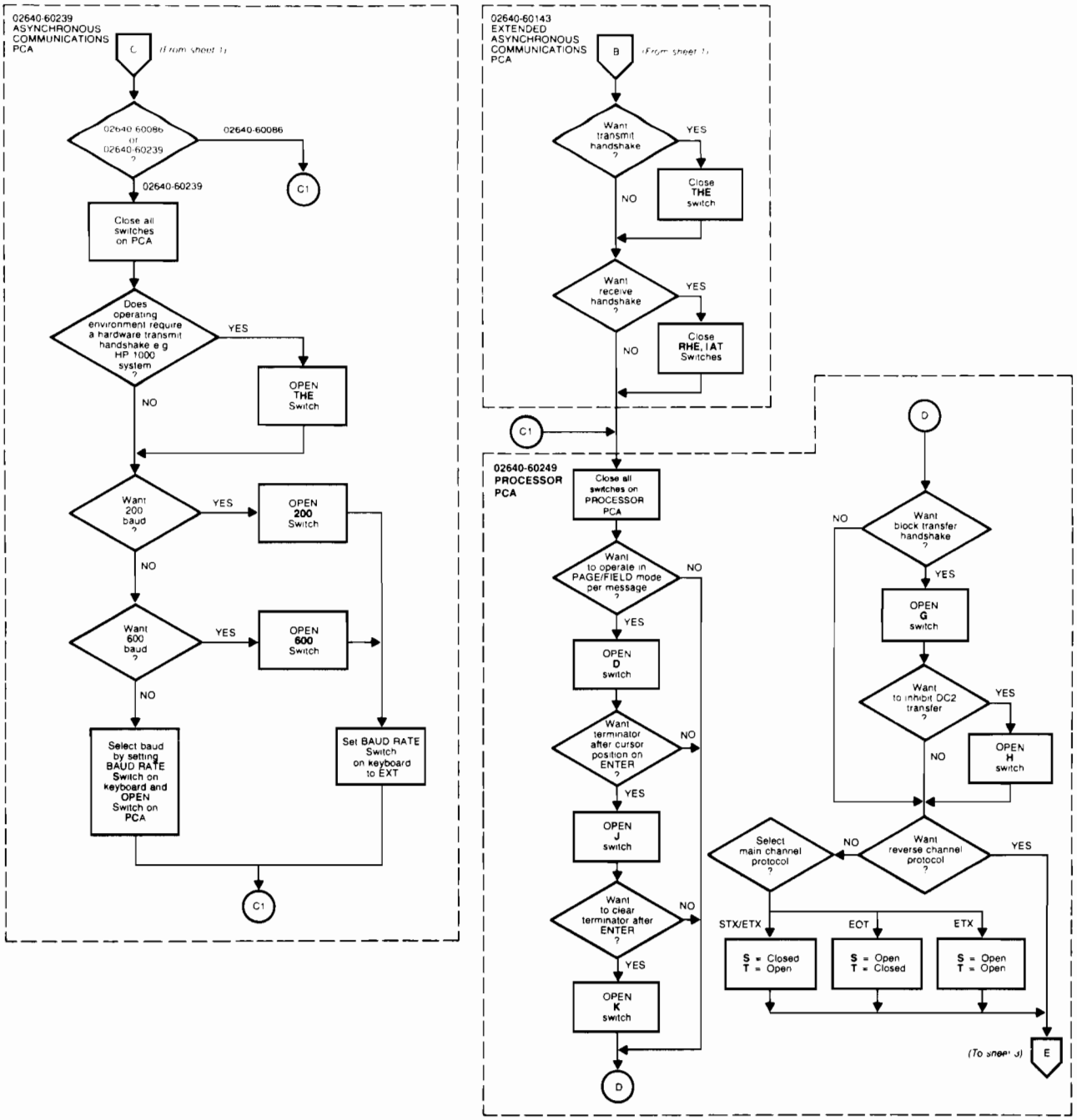


Figure 12-4. Data Communications Configuration Flowchart (Sheet 2 of 4)

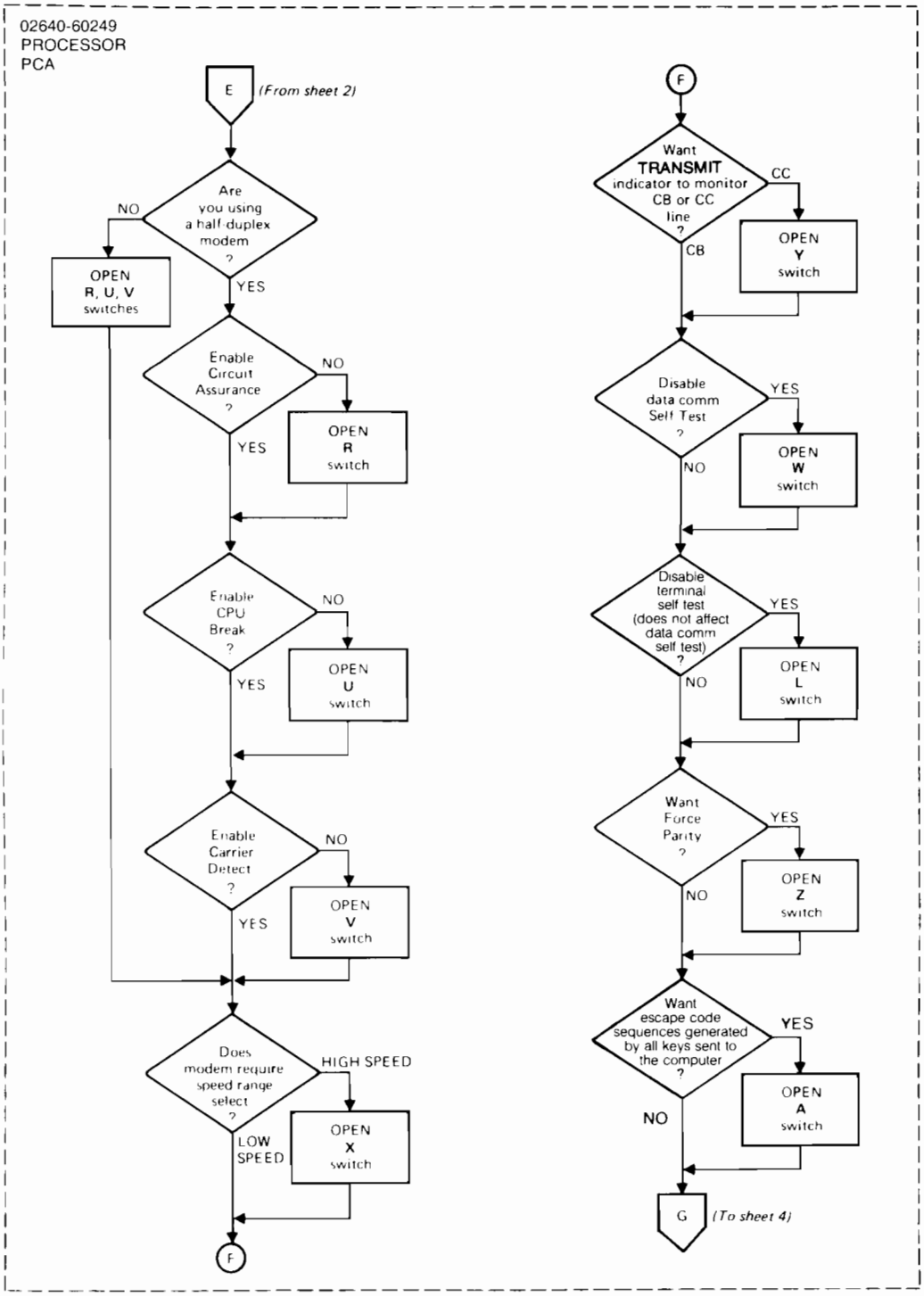
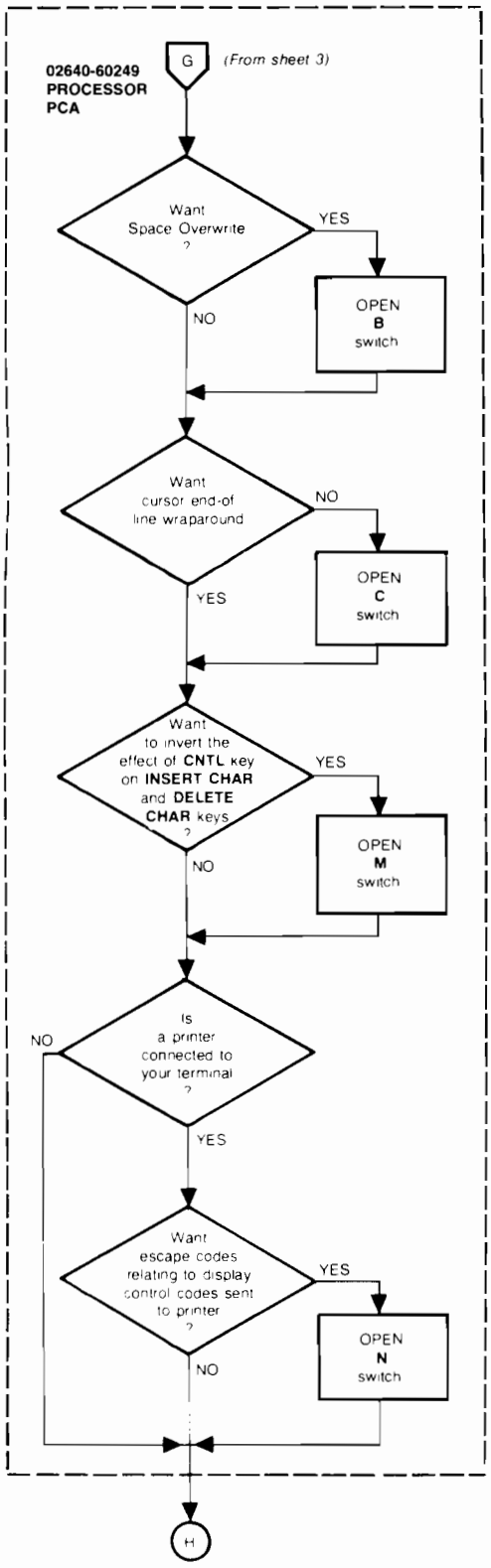
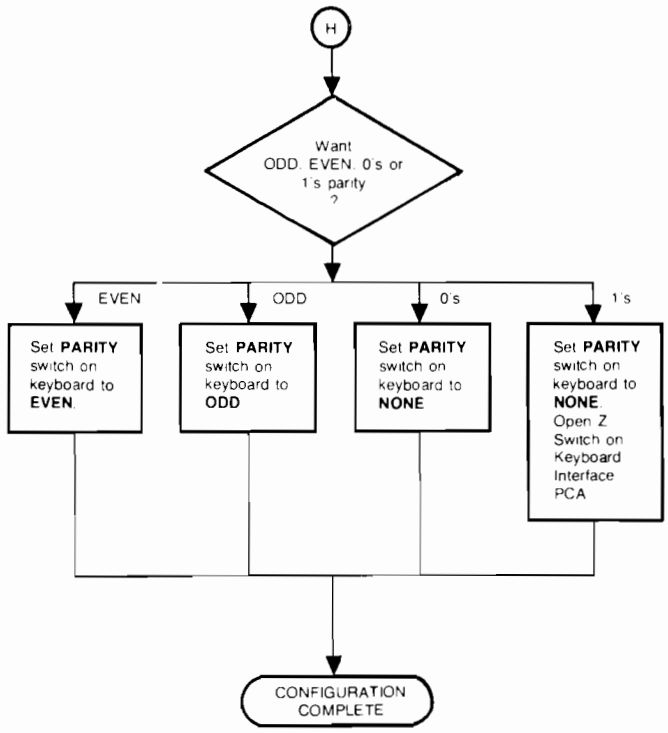


Figure 12-4. Data Communications Configuration Flowchart (Sheet 3 of 4)





BAUD RATE SELECTIONS

X = Available Split Rate C = Switch Closed O = Switch Open		TRANSMIT BAUD RATE										BAUD RATE SWITCH SETTING
		EXT	CUSTOM	110	150	300	1200	2400	4800	9600	134.5	
	EXT	X	X		X	X	X	X	X	X		EXT
	CUSTOM		X		X	X	X	X	X	X		EXT (See Notes)
	110		X	X								110
	150		X		X	X	X	X	X	X		150
	300		X		X	X	X	X	X	X		300
	1200		X		X	X	X	X	X	X		1200
	2400		X		X	X	X	X	X	X		2400
	4800		X		X	X	X	X	X	X		
	9600		X		X	X	X	X	X	X		
	134.5		X								X	
S2S0 Setting		O	C	O	C	O	C	O	C	O	O	
S2S1 Setting		O	C	O	O	O	C	C	O	C	O	
S2S2 Setting		O	C	O	C	C	O	O	O	C	O	

NOTES

The S2CBE switch must be set to its closed position for custom receive baud rates and keyboard BAUD RATE switch must be set to EXT.

When switches S2S0, S2S1, and S2S2 are all set to their open positions, the transmit baud rate will equal the receive baud rate.

Figure 12-4. Data Communications Configuration Flowchart (Sheet 4 of 4)

HP-IB/ROM PCA

Configuration of the HP-IB/ROM PCA consists of selection of the PCA address of the HP-IB/ROM PCA within the terminal, selection of the HP-IB address of the terminal within the HP-IB network, selection of whether or not the terminal is to be the system controller (SC), and setting the unused switches to the open position.

Operational Considerations

Any terminal in the network can be selected as the System Controller. This is accomplished by setting switch SC on the terminal's HP-IB/ROM PCA to the "open" position. Only one terminal, however, should be assigned that role. An HP-IB network cannot function if it includes two or more System Controllers. The system controller should be the terminal with the lowest HP-IB address.

At any given time one of the terminals in the network is designated as the "Controller in Charge" (or CIC). Note that this is completely independent of the System Controller function. It merely means that the particular terminal was the one which most recently issued a command over the HP-IB. Whenever a terminal is currently the CIC its red LED (above the f4 key) is lit.

If power is removed from the terminal which is currently designated as the CIC, the entire network becomes inoperative. To recover from such an event, you must do a hard reset on the System Controller terminal (note that the terminal which was powered off need not be turned back on unless it also happens to be the System Controller terminal.)

If power is removed from any terminal which is not currently designated as the CIC, the network is unaffected.

CAUTION

Doing a hard reset on the CIC halts any operation currently in progress. To resume operation you must then do a hard reset on the System Controller terminal.

Doing a hard reset on the System Controller terminal re-initializes the HP/IB network.

To configure the HP-IB/ROM PCA, set the configuration switches on the PCA using table 12-5 as a guide. Figure 12-5 locates the configuration switches on the PCA.

Sample Networks

Figure 12-6 illustrates an HP-IB network containing only two devices: a terminal and an HP 7310A Printer.

The terminal is designated as the System Controller with an HP-IB device address of 29. To configure the terminal in that manner, switches B4 through B0 and SC on the HP-IB/ROM PCA are set as follows:

LSB	B0:	Open	
	B1:	Closed	These switch settings specify the
	B2:	Open	terminal's HP-IB device address as
	B3:	Open	29.
MSB	B4:	Open	
	SC:	Open	This switch setting indicates that the terminal is the System Controller.

Since the printer is located adjacent to the terminal in this sample network, the standard 2 meter HP-IB interconnecting cable is adequate and the HP-IB/ROM PCA is set to "0" additional device loads.

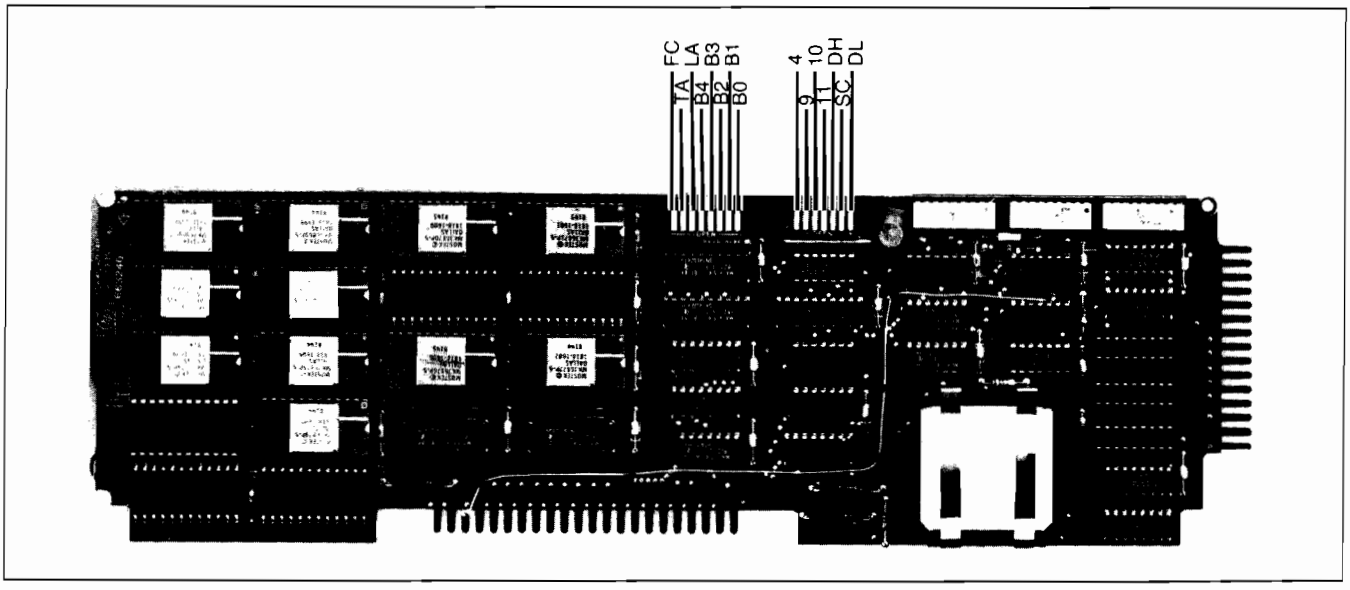


Figure 12-5. HP-IB/ROM PCA Configuration Switches

Table 12-5. HP-IB/ROM PCA

SWITCH(ES)	SETTING(S)
A4, A11, A10, A9	These four switches specify the PCA module address and must be set as follows: A4 Closed, A11 Open, A10 Closed, A9 Closed
FC, and TA	These two switches are reserved for future use and should be set to the closed position.
LA	This switch should be set to the open position to indicate the presence of the HP-IB/ROM PCA.
B4 through B0	These five switches specify the HP-IB address of your terminal. The device addresses 0-7 are reserved for peripheral devices (plotters and printers) while the device addresses 8-29 may be assigned to terminals. To achieve the most efficient operation, it is recommended that the addresses for a terminal cluster start at 29 and work their way down sequentially. The terminal which is assigned as the System Controller of the HP-IB (see switch SC below) must, however, be assigned the lowest device address of all the terminals in the shared peripheral configuration. The switch settings for the various terminal HP-IB addresses are as follows:
DH, DL	These two switches should be set to the open position.

SWITCH(ES)	SETTING(S)

Decimal Address	B4	B3	B2	B1	B0
8	C	O	C	C	C
9	C	O	C	C	O
10	C	O	C	O	C
11	C	O	C	O	O
12	C	O	O	C	C
13	C	O	O	C	O
14	C	O	O	O	C
15	C	O	O	O	O
16	O	C	C	C	C
17	O	C	C	C	O
18	O	C	C	O	C
19	O	C	C	O	O
20	O	C	O	C	C
21	O	C	O	C	O
22	O	C	O	O	C
23	O	C	O	O	O
24	O	O	C	C	C
25	O	O	C	C	O
26	O	O	C	O	C
27	O	O	C	O	O
28	O	O	O	C	C
29	O	O	O	C	O

where O = switch open
C = switch closed

SC	This switch determines whether or not your terminal will be the System Controller of the HP-IB (remember that the System Controller must have the lowest device address of all the terminals in the shared peripheral configuration). One, and only one, terminal in the shared peripheral configuration must be assigned as the System Controller. Open = System Controller

The printer is assigned HP-IB device address 5 by setting the address switches A1 through A5 on the back panel as follows:

LSB A1: 1 (on)
A2: 0 (off)
A3: 1 (on)
A4: 0 (off)
MSB A5: 0 (off)

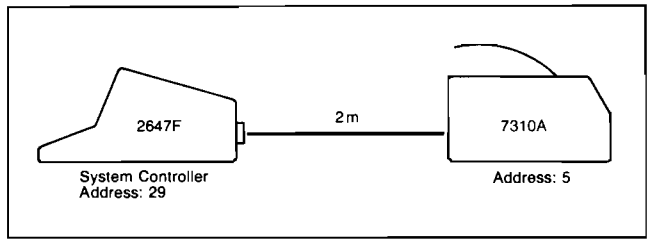


Figure 12-6. Terminal-to-Printer Network

Once the network is established as described above, you may specify the printer as the output device for printing by specifying HP-IB #5 as the "destination" or "to" device, where the "5" specifies the plotter's HP-IB device address.

Figure 12-7 illustrates an HP-IB network containing four devices: one HP 2647F terminal, one HP 2647A terminal, one HP 7245A Plotter/Printer, and one HP 2631A/G Printer.

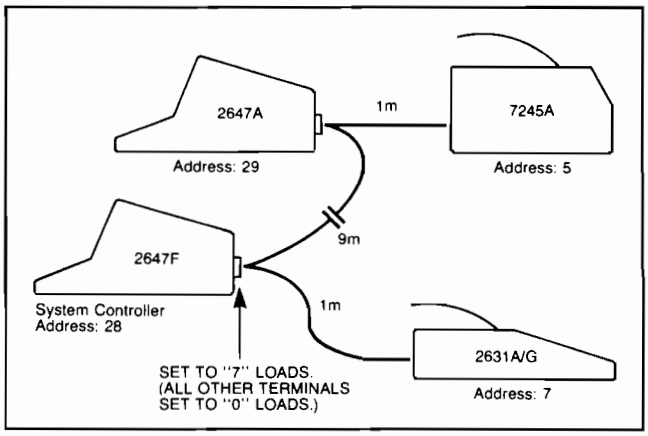


Figure 12-7. Shared Plotter/Line Printer Network

The terminals are assigned HP-IB device addresses 29 and 28, the latter also being designated as the System Controller. To configure the terminals in that manner, switches B4 through B0 and SC on their HP-IB Interface PCAs are set as follows:

LSB	B0:	Open	
	B1:	Closed	These switch settings specify the
	B2:	Open	2647F terminal's HP-IB device
	B3:	Open	address as 28.
MSB	B4:	Open	
	SC:	Open	This switch setting indicates that the terminal is the System Controller.

LSB	B0:	Closed	
	B1:	Closed	These switch settings specify the
	B2:	Open	2647A terminal's HP-IB device
	B3:	Open	address as 29.
MSB	B4:	Open	
	SC:	Closed	This switch setting indicates that the terminal is not the System Controller.

The HP 7245A Plotter/Printer is assigned HP-IB device address 5 by setting the address switches A1 through A5 on the back panel as follows:

LSB	A1:	1
	A2:	0
	A3:	1
	A4:	0
MSB	A5:	0

NOTE: Address 5 is assigned to the Plotter function of the HP 7245A; address 6 is automatically assigned to the Printer function of the HP 7245A.

The HP 2631A/G Printer is assigned HP-IB device address 7 by setting the address switches 1 through 5 on the back panel as follows:

LSB	5:	1 (closed)
	4:	1 (closed)
	3:	1 (closed)
	2:	0 (open)
MSB	1:	0 (open)

Note that switches 6 and 7 on the back panel of the HP 2631A/G should both be set to the open position.

The HP 2631A/G Printer is one of those devices which limits the overall amount of HP-IB interconnecting cable to an average of one meter per device load. With four actual devices plus seven additional device loads simulated by the HP-IB Interface Adapter of the System Controller terminal, we can therefore have up to 11 meters of HP-IB interconnecting cable in our sample network.

Figures 12-8 and 12-9 illustrate larger HP-IB networks.

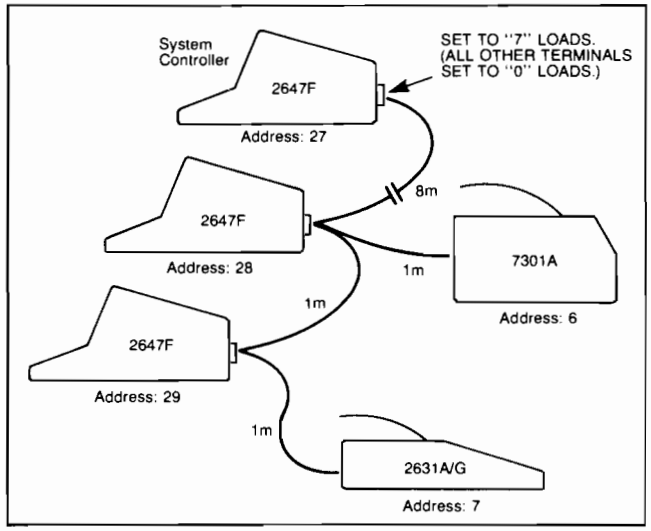


Figure 12-8. 3-Terminal, Shared Printer Network

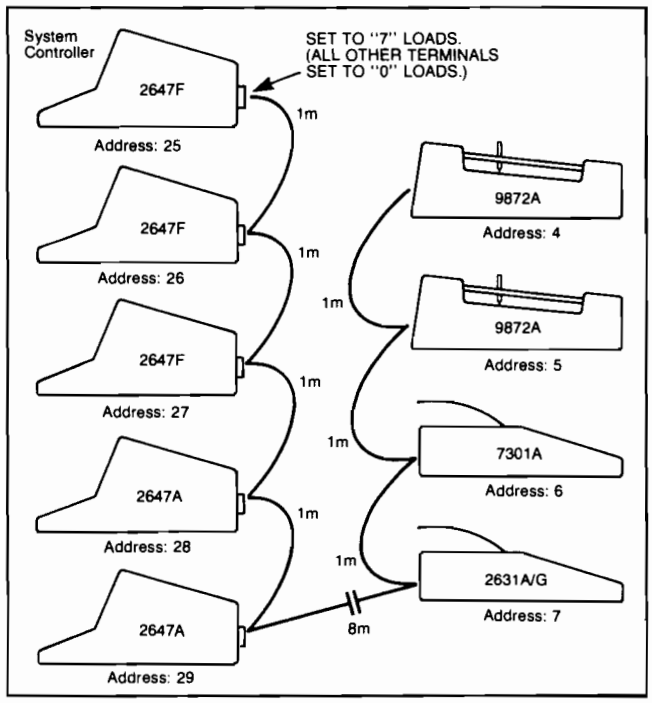


Figure 12-9. 5-Terminal, Multiple Plotter/Printer Network

DISPLAY PCAs CONFIGURATION

Configuration of the terminals display circuits is done using switches on the Display Memory/DMA PCA and the Display Timing/Control PCA (see figures 12-10 and 12-11). The switches are set as follows (a switch setting of "1" indicates the switch is closed; a "0" setting indicates an open switch):

DISPLAY MEMORY/DMA PCA 02640-60250

SWITCH	SETTING
1	1
2	1
3	1
4	1

The switches on the Display Timing/Control PCA are used to select the character set or pattern for display on the screen. Switches A1, A2, A3, U1, U2, and U3 are set to the following positions:

SWITCH	SETTNG
A1	0
A2	1
A3	1
U1	0
U2	0
U3	0

Switches C1, C2, and C3 assign the Math, Line Drawing, and Large Character sets to the character set designations A, B, and C. This enables use of the ↰) escape sequence to select set A, B, or C as the active character set. (Refer to Appendix D for further information on the ↰) escape sequence). The following table shows the relationship between the switch settings and the character set assignments.

SWITCH			CHARACTER SET		
C1	C2	C3	A	B	C
0	0	0	Math	Line	Large
0	0	1	Math	Large	Line
0	1	0	Line	Math	Large
0	1	1	Line	Large	Math
1	0	0	Large	Math	Line
1	0	1	Large	Line	Math
1	1	0	Math	Line	Large
1	1	1	Math	Line	Large

The functions of the remaining switches are shown below.

SWITCH	POSITION	FUNCTION
U3	0	Displays full screen crosshatch in Display Test mode.
	1	Displays full screen at (@) and delete (■) characters.
DT	0	Display Test mode on.
	1	Display Test mode off.
IV	0	Normal display mode.
	1	Full screen inverse video.
HZ	0	60 Hz power source selected.
	1	50 Hz power source selected.

NOTE: The HZ switch setting is only sampled at power-on time.

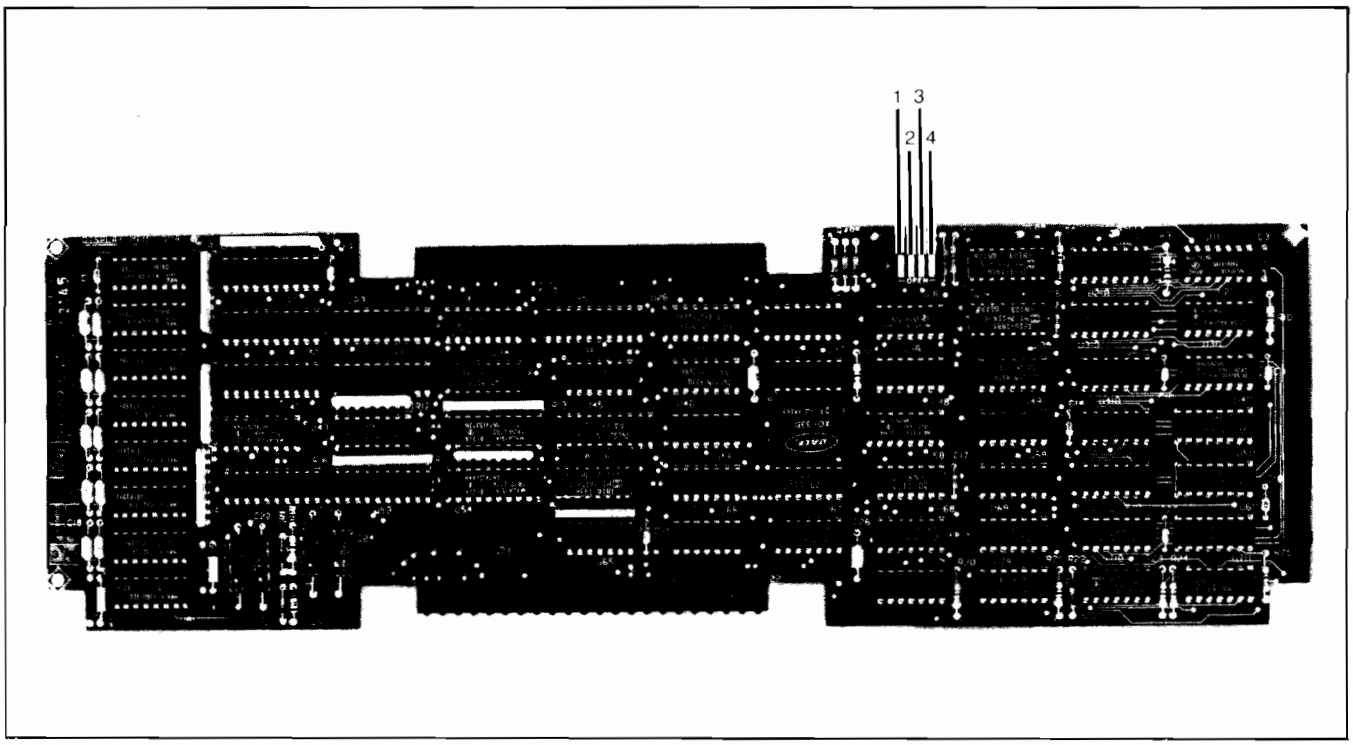


Figure 12-10. Display Memory/DMA PCA
Configuration Switches

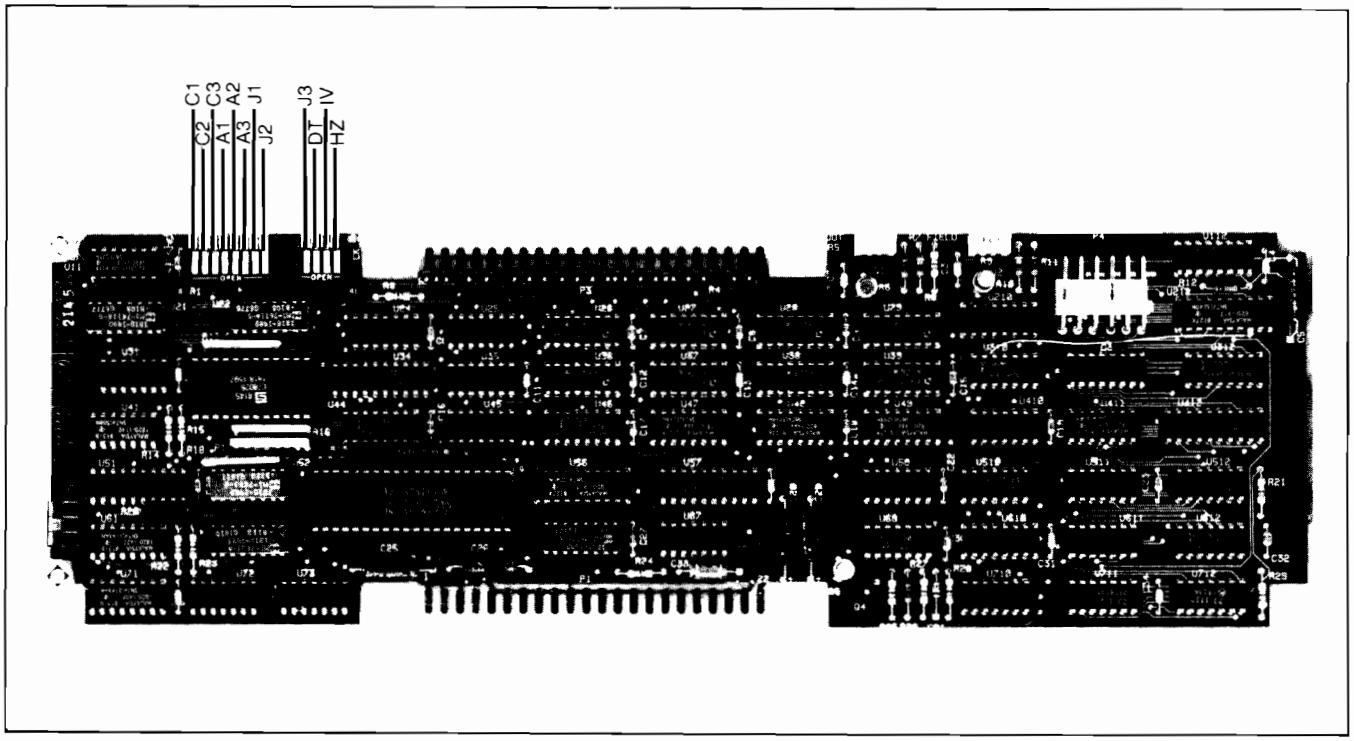


Figure 12-11. Display Timing/Control PCA
Configuration Switches

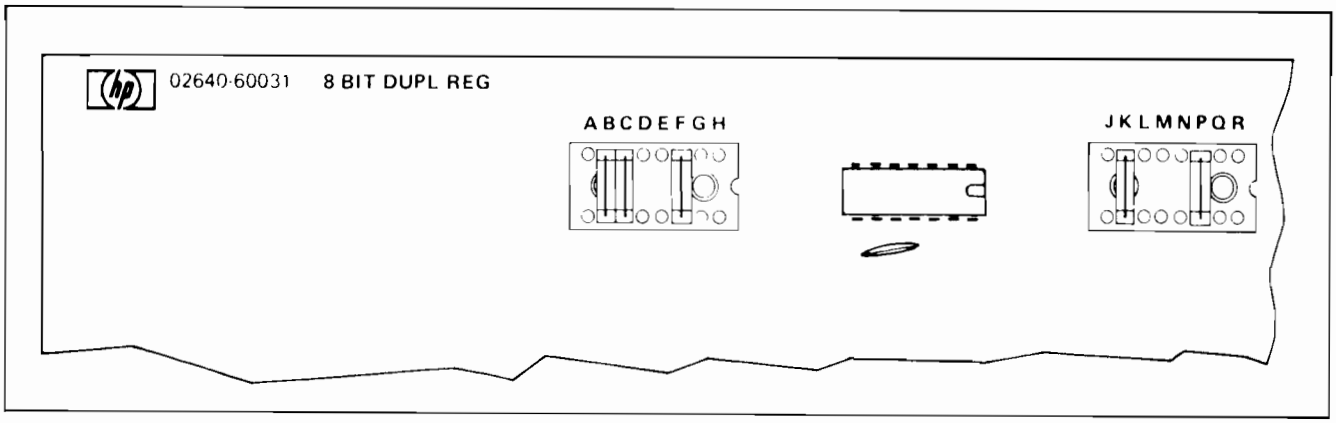


Figure 12-12. Terminal Duplex Register PCA Jumper Configuration

HP 13238A TERMINAL DUPLEX REGISTER CONFIGURATION

Configuration of the Terminal Duplex Register PCA consists of inserting five jumpers in the PCA jumper positions indicated in figure 12-12.

HP 13250B SERIAL PRINTER INTERFACE

Refer to the manual "HP 13250A/B Serial Printer/Extended Asynchronous Data Communications Interface Operating Manual", part no. 13250-90004, for configuration instructions on the 13250A/B accessory.

HP 13254A VIDEO INTERFACE

Refer to the manual "13254A Video Interface Accessory Operating and Service Manual", part no. 13254-90001, for configuration instructions.

HP 13260B EXTENDED ASYNCHRONOUS DATA COMMUNICATIONS INTERFACE

Refer to the manual "HP 13250A/B Serial Printer/Extended Asynchronous Data Communications Interface Operating Manual", part no. 13250-90004, for configuration instructions.

SELF-TEST

SECTION 13

INTRODUCTION

The terminal tests itself. Should you suspect a malfunction while operating the terminal, you can perform the SELF-TEST function to checkout the terminal. Also, after installing any accessory, the terminal's self-test function should be performed to ensure that the terminal is functioning properly. The terminal has several self-tests:

- Basic Self-Test
- Disc Self-Test
- Data Communications Self-Test
- HP-IB Self-Test
- Terminal-to-Terminal Loop-Back Test

BASIC SELF-TEST

Pressing [TEST] checks out the terminal, except for the disc drives and the data communications. In addition, basic self-test does not test the HP-IB. The following is performed when the [TEST] key is pressed (also see the flowchart in figure 13-2):

NOTE: The test pattern cannot be recorded because of embedded Record Separators (␞).

- The light-emitting diodes (indicators on the keyboard are turned on for the duration of the test as an indication that the power supply and microprocessor board are functioning.

- A checksum test is done on the read-only memory (ROM). This verifies that the firmware is working properly. An error here causes a bad ROM message to be displayed. (See flowchart, figure 13-2.) Note that this checksum test is also performed as part of the power-on sequence.

- A RAM TEST is performed on all random access memory (RAM). The message "TESTING RAM" is displayed. Then the checkerboard test is performed. An error here causes a RAM ERROR message. (See flowchart, figure 13-2.)

- The bell is beeped indicating success up to this point. If the [TEST] key is held down to cause the self-test to be repeated, the bell beeps only for the first self-test.

- The entire character set contained in the terminal is displayed.

- A line of characters, @ABCDEFGHIJKLMNO, is displayed. The Underline, Half-Bright, and Blinking will be displayed with Inverse Video in all of the possible Display Enhancement combinations by this line of characters.

- The 14 bytes of status information are displayed. (See the "Status" section for an explanation of the status bytes.)

Generally, if the terminal beeps and the display shows a pattern similar to those shown in figure 13-1 then the terminal is functioning properly (only those character sets actually present in the terminal will be displayed in the test pattern and consequently the actual test pattern displayed will be dependent on which features are present in each terminal.

[RESET TERMINAL] must be pressed to resume operation if any error occurred. However, the station's operation will not be reliable if the Self-Test failed.

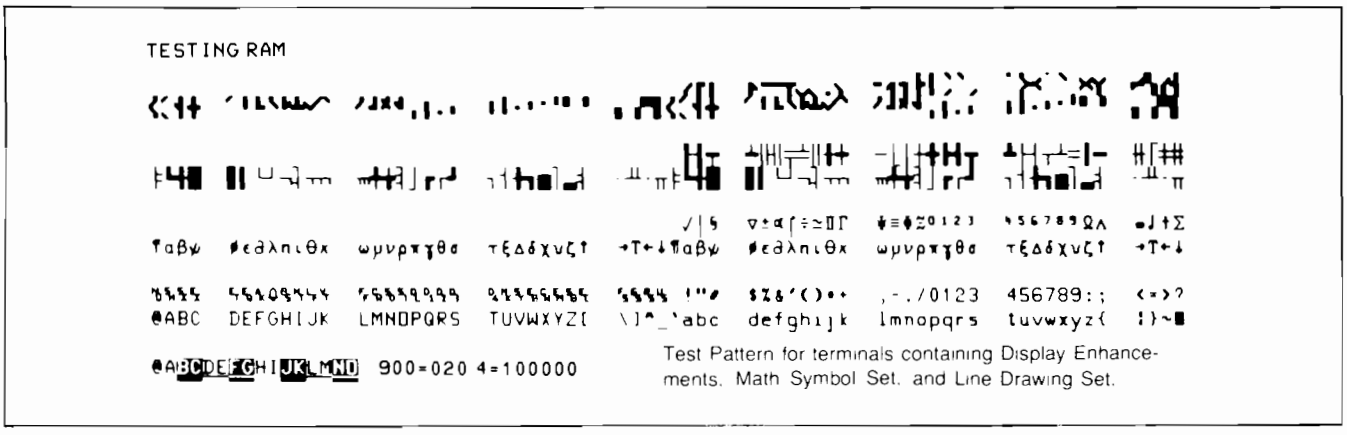


Figure 13-1. Basic Self-Test Patterns

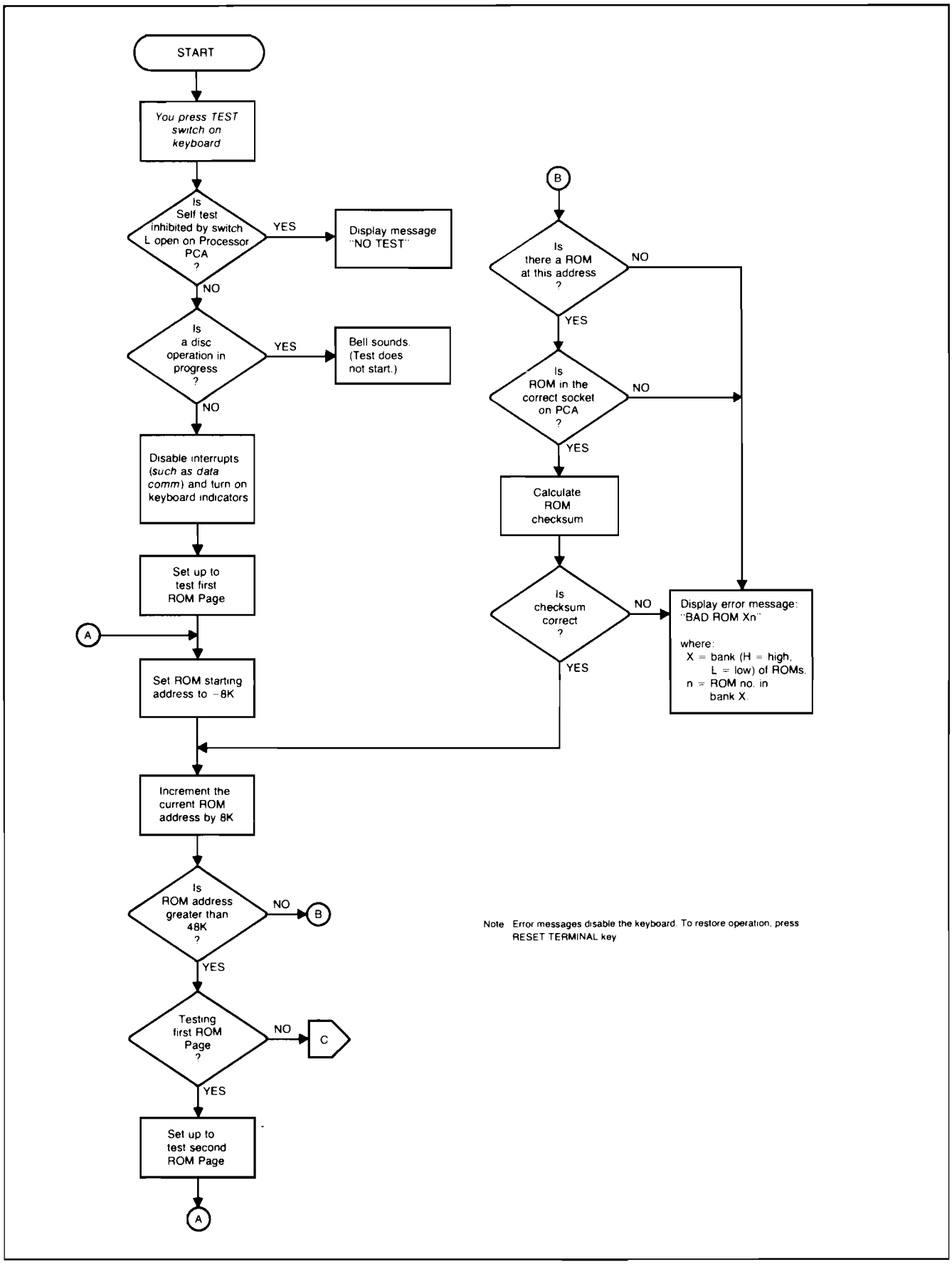


Figure 13-2. Basic Terminal Self-Test Flowchart (Sheet 1 of 4)

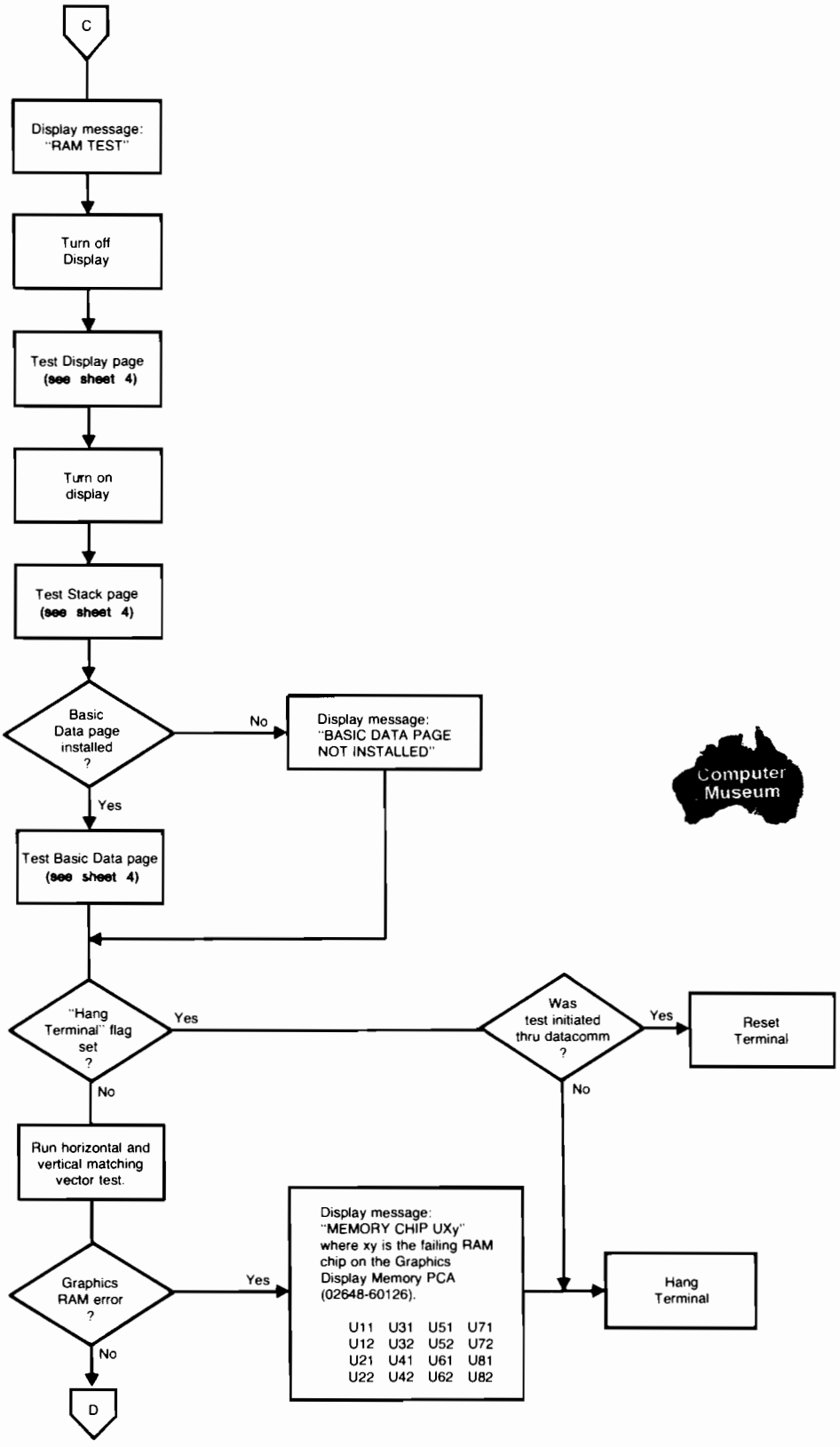
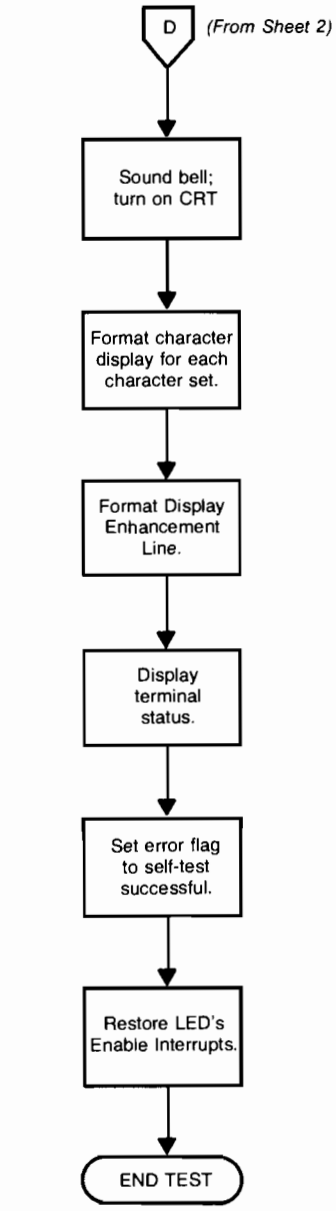


Figure 13-2. Basic Terminal Self-Test Flowchart (Sheet 2 of 4)



Note: Error messages disable the keyboard. To restore operation, press RESET TERMINAL key.

Figure 13-2. Basic Terminal Self-Test Flowchart (Sheet 3 of 4)

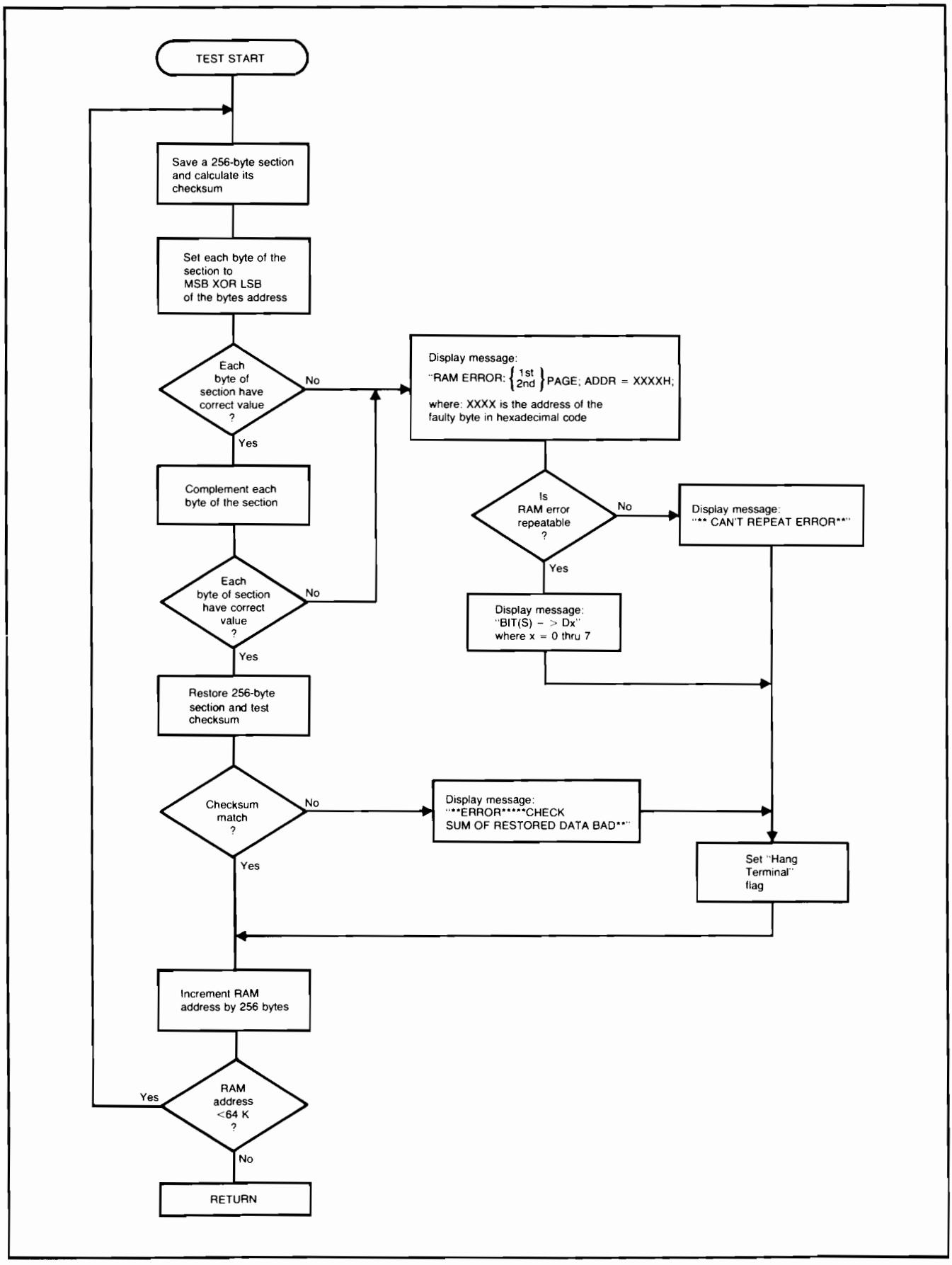


Figure 13-2. Basic Terminal Self-Test Flowchart (Sheet 4 of 4)

DISC SELF-TEST

The disc self-test is performed with a write-enabled diskette mounted in each disc drive. Make sure that any data on these diskettes need not be saved. This test checks the following disc architecture and operations:

- Controller PCA
- Drive PCA
- Disc Drive Unit
- Writing
- Reading

Performing the Disc Self-Test. Pressing the COMMAND key, and keys next, next, next, TEST, DISCS, and then the RETURN key initiates the following:

- A non-destructive test is performed on the Controller PCA at Power-On.
- A destructive (write/read) test is initiated on the diskette(s) mounted in the disc drive(s). The write test reformats and writes the worst case data pattern on a pre-selected cylinder (32).

Following the execution of self-test, the terminal displays a message on the screen to indicate if the disc has successfully completed the test or if an error occurred. Messages that may appear are:

```
Disc self test OK, No disc controller attached
Disc self test Fail, Unit is <u>, Test is <t>,
  Subtest is <s>, Head is <h>
Disc self test fail, automount disable
```

where:

<u> is the selected disc drive unit when the failure occurred.

<t> indicates which test failed.

<s> indicates which subtest of the current test failed.

<h> indicates which head the test failed on.

Possible failing tests and their explanations are given in table 13-1.

If a hardware failure occurred during self-test, the relability of the terminal cannot be assured. If any error occurred, try replacing the diskette and running the self-test again to make sure that the error is a hardware malfunction. Servicing procedures are contained in the *Service Manual.*

DATA COMMUNICATIONS SELF-TEST

This self-test checks the data communications PCA and the associated network cabling. Test connectors are used with the self-test function to provide signal loop-back while the internal diagnostic is being run. A description of the test connectors is provided in table 13-2. To run the self-test, follow the instructions in table 13-3. A flowchart of the self-test is contained in figure 13-3.

HP-IB SELF-TEST

The HP-IB self-test tests the PHI IC chip. To initiate the test, type `TEST HP-IB` in the command channel (you can also do this using the TEST and HP IB softkeys) and then press RETURN.

If the HP-IB test fails, the following message is displayed:

```
HP-IB TEST FAIL, ERROR=x
```

where x is a number from 0 to 2 whose meaning is as follows:

0 indicates no HP-IB/ROM PCA is installed (or the PCA, if present, is strapped incorrectly).

1 or 2 indicates a PHI-IC chip error.

If the test is successful, the following message is displayed:

```
                                        YES     YES
HP-IB TEST OK,ADDR=xx,SYSCTL=    ,CIC=
                                        NO      NO
```

The above message indicates the HP-IB/ROM PCAs' current strapping and functional configuration. **xx** is an address, from 8 to 29, which uniquely identifies your terminal's HP-IB/ROM PCA and differentiates it from all other HP-IB devices connected to the HP-IB bus. The SYSCTL parameter indicates whether or not your terminal is configured as the System Controller of the HP-IB. The CIC parameter indicates whether or not your terminal is currently the Controller in Charge (CIC) of the HP-IB bus. CIC=YES merely means that your terminal was the terminal which most recently issued a command over the HP-IB.

Table 13-1. Disc Self-Test Description

TEST ITEM	TEST NO.	SUBTEXT NO.	TEST	EXPLANATION
Controller PCA	15	—	Power-On	Controller cannot be initialized
	15	—	LED Test	LED(s) changed from on to off
	4	—	ROM Test	CRC Check
	5	—	RAM 1 Test	Data Pattern LSB
	6	—	RAM 2 Test	Data Pattern MSB
	7	—	RAM 3 Test	Address Line Error
	2	0	FDC Test1	Floppy Disc Controller Time-Out Error
	9	4	FDC Test2	Floppy Disc Controller Interrupt Error
	8	6	Write Circuit Test	(1) Data pulse width = 0 or > 9μs. (2) 9us one-shot multivibrator circuit is defective. (3) Controller cable is broken or not connected to drive.
	10	7	I/O Test	I/O and/or Handshake Registers may be defective
Drive PCA Test (Note 1)	11	3	Seek Test1	(1) After recalibration, track 0 indicator was not detected. (2) Tract 0 indicator circuit is defective. (3) Stepper motor did not move the acutator towards track 0. (4) Controller cable is broken or not connected.
	11	5	Seek Test2	(1) Track 0 indicator circuit is defective. (2) Stepper motor did not move the acutator towards track 34.
	11	6	Seek Test3	(1) Track 0 indicator circuit is defective. (2) Stepper motor did not move the acutator towards track 0.
	11	7	No Drive	(1) Drive ID circuit is defective. (2) Controller cable is not connected to drive or is broken.
	11	8	Write/Read	(1) Data pulse width = 0 or > 9 μs. (2) Defective write/read circuit. (3) Defective 9 μs one-shot multivibrator circuit. (4) Controller IC is defective on drive PCA. (5) Controller cable is not connected to drive or is broken.
	11	9	9 μs Test	(1) Defective 9 μs one-shot multivibrator circuit. (2) Controller cable is not connected to drive or is broken.

Table 13-1. Disc Self-Test Description (Continued)

TESTED ITEM	TEST NO.	SUBTEST NO.	TEST	EXPLANATION
Drive Test (Note 2)	12	0	No Index Mark	(1) Defective index circuit. (2) Defective photo detector circuit or LED indicator. (3) Motor is not spinning. (4) Door is not closed. (5) Controller IC is defective.
	12	1	Speed Test1	Disc spinning faster than specified limits.
	12	2	Speed Test2	Disc spinning slower than specified limits.
	12	6	Missing Pulse	(1) Defective missing one-shot circuit (2) Defective index circuit.
Write Test (Note 3)	13	1	No Diskette	(1) No diskette inserted in drive or door not closed on drive. (2) Defective missing pulse one-shot circuit.
	13	2	Write Protected	(1) Defective write protect circuit (2) Diskette is write protected.
	13	3	Underrun	(1) Controller IC did not receive data before 30 us. (2) Defective timing circuit.
Read Test (Note 4)	14	0	Not Ready	(1) Defective disc change circuit (2) Diskette is missing from drive.
	14	2	No ID Field	(1) Diskette is defective (2) Unable to format diskette due to defective write circuit. (3) Defective controller IC.
	14	3	Wrong ID	Seek failure caused by wrong track.
	14	5	CRC Error	(1) Defective controller IC. (2) Phase lock loop error caused by defective read circuit.
	14	6	Overrun	(1) Overrun occurred from lost data in the current read attempt. (2) Timing circuit is defective.

Notes:

1. A failure in the Drive PCA Test indicates a problem associated with the Drive PCA, or controller/drive logic.
2. A failure in the Drive Test indicates a problem with the motor speed, stepper motor, or the photo detection circuit on the disc drive unit.
3. A failure in the Write Test indicates a problem with recording data or formatting the diskette in the indicated drive.
4. A failure in the Read Test indicates a problem in reading data from the diskette. This may result from a failure to write or read correctly or from bad media.

Table 13-2. Data Communications Self-Test Connectors

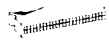
SELF-TEST CONNECTORS	HP PART NO.	USED FOR
	02645-60002	Checks RS232 circuits on 13260A, B accessory PCA's. (Does not check multipoint circuits on PCA; use 02645-60004 test connector below.) This connector is supplied with the terminal.
	02645-60035	Checks current loop circuits on 13260B accessory PCA. This connector is supplied with the 13232F Current Loop Cable.
	02645-60004	Provides loop-back of RS232 signals at RS232 connector end of cable. Used during self-test of multipoint configurations.

Table 13-3. Point-to-Point Data Communications Self-Test Procedure

STEP 1.

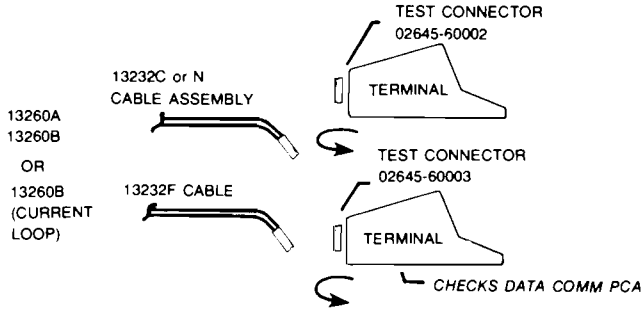
a. Ensure power is off, and disconnect cable on the data communications PCA.

b. Connect PCA Test Connector, part no. 02645-60002, to data communications PCA.

c. Turn on power, type TEST DATACOMM in the command channel (you can also do this using the TEST and DATACOMM softkeys), and then press [redacted].

d. Refer to data comm self-test flowcharts for diagnosing possible error messages.

e. If operating in current loop, turn power off and use test connector part no. 02645-60003 to connect to the 13260B Data Communications PCA. Turn on power, and type characters on the keyboard. The characters should be echoed back (two characters displayed if the terminal is set for Half Duplex). This verifies proper operation of current loop send and receive circuits.



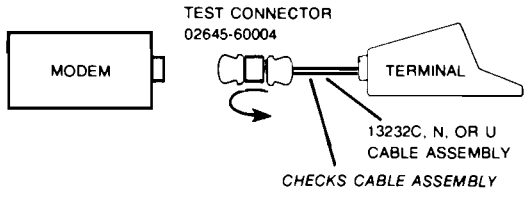
STEP 2.

a. Turn off power, and connect 13232C or N Cable Assembly to 13260A or B data communications PCA. (If operating in current loop, connect 13232F cable to 13260B data communications PCA.)

b. Connect RS232 Test Connector, part no. 02645-60004, to RS232 connector on 13232C or N cable.

c. Turn on power, set the terminal to REMOTE, type TEST DATACOMM in the command channel (you can also do this using the TEST and DATACOMM softkeys), and then press [redacted].

d. Refer to data comm self-test flowcharts for diagnosing possible error messages.



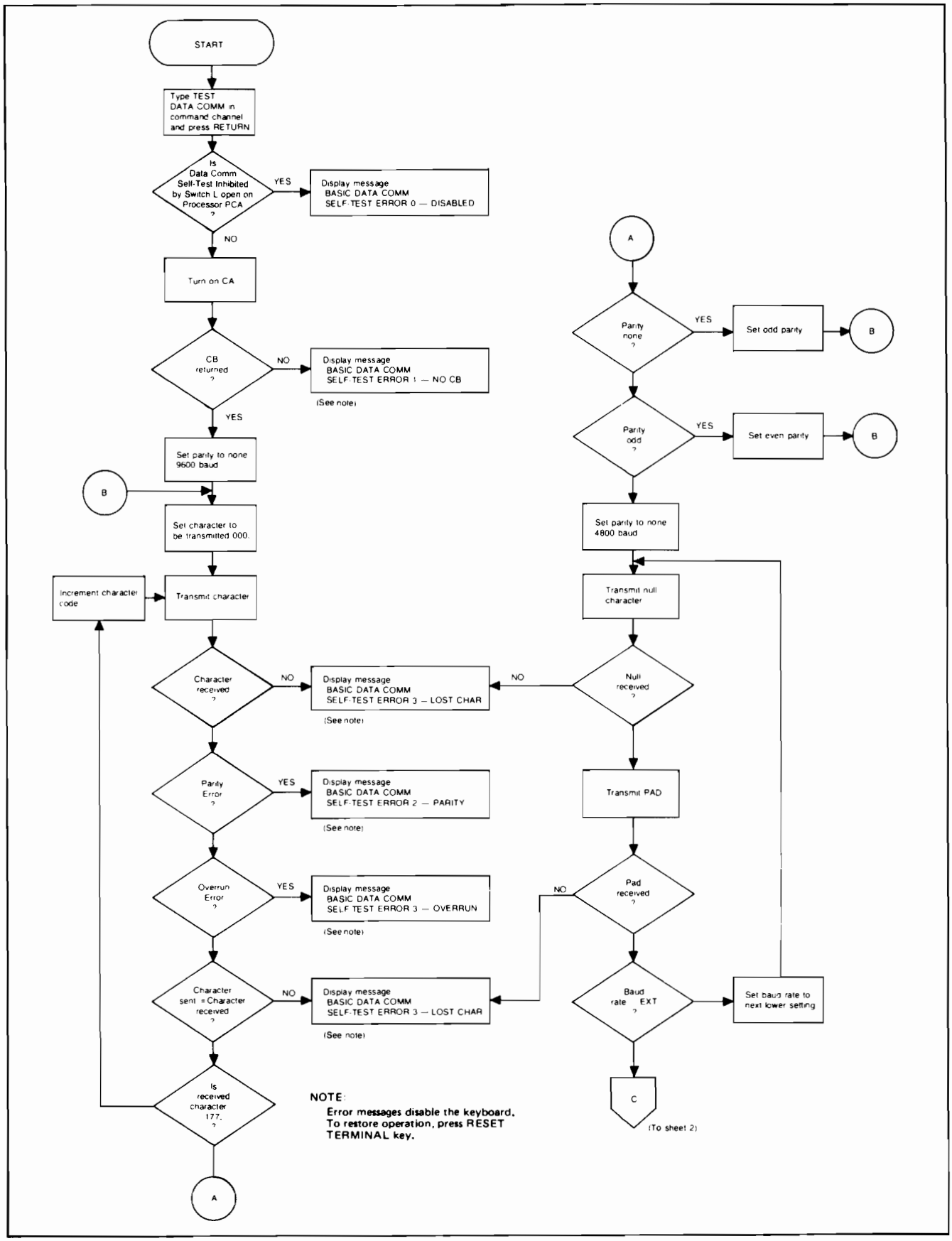


Figure 13-3. Basic Data Comm Self-Test Flowchart (Sheet 1 of 2)

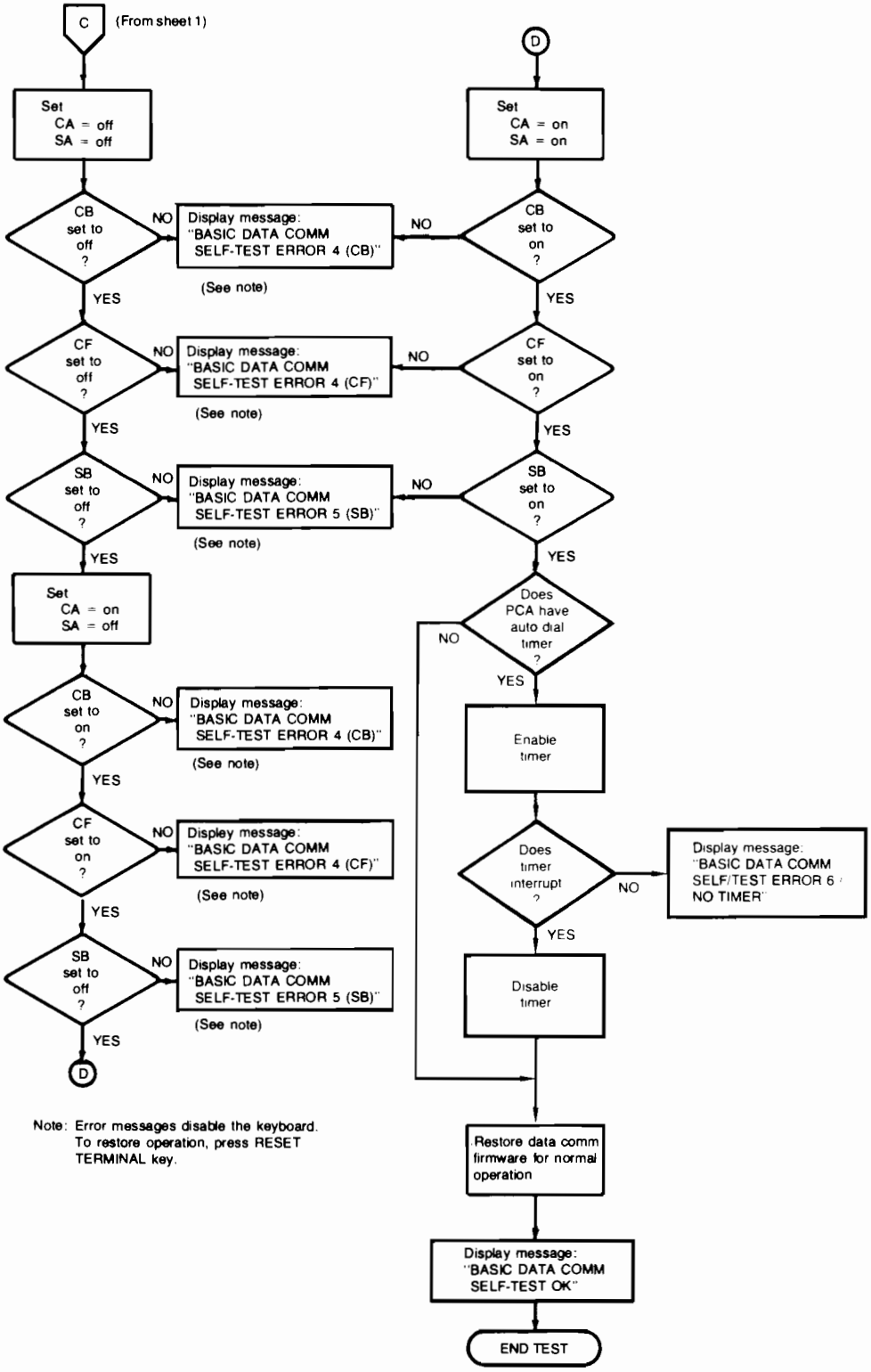


Figure 13-3. Basic Data Comm Self-Test Flowchart (Sheet 2 of 2)

In order for the HP-IB self-test to be performed, your terminal must be:

1. Both the System Controller and the CIC; or
2. Neither the System Controller nor the CIC.

If neither of the above is true (that is, if SYSCTL=NO,CIC=YES or SYSCTL=YES,CIC=NO) then the test cannot be performed and the following message is displayed:

```
HP-IB TEST NOT ATTEMPTED,ADDR=xx,

        YES      YES
SYSCTL=    ,CIC=
        NO       NO
```

In such a case, it is most convenient to manipulate the CIC function (manipulating whether or not your terminal is the System Controller requires reconfiguring the entire HP-IB configuration). To make your terminal the CIC, type `SHOW TIME` in the command channel and then press RETURN. To make it so that your terminal is not the CIC, type `SHOW TIME` in the command channel (followed by RETURN) of another terminal in the configuration. Then retry the HP-IB self-test.

TERMINAL-TO-TERMINAL LOOP-BACK TEST

To test the communication capabilities between two terminals in an HP-IB network, enter `TEST TERMINAL#x` in the command channel, where `x` is the HP-IB device address of the target terminal, and then press RETURN. This initiates a loop-back test in which a 256-byte data block is sent from your terminal to the target terminal and is then sent back to your terminal for comparison.

If the test fails, then the following message is displayed:

```
HP-IB TE#x TO TE#y TEST FAILED, ERROR:z
```

where: `x` is the HP-IB device address of your terminal.

`y` is the HP-IB device address of the target terminal.

`z` is the error number, as follows:

- 1 indicates no HP-IB interface PCA
- 2 indicates no control of HP-IB
- 3 indicates target terminal did not receive entire data block
- 4 indicates target terminal did not send back entire data block
- 5 indicates returned data block did not match the one originally transmitted.

ERROR MESSAGES

APPENDIX A

INTRODUCTION

Errors detected by the terminal's file system or user interface have decimal error codes within the range 257-1023 (inclusive). Below is a list of all the file system/user interface error codes.

FILE SYSTEM INTRINSIC ERROR MESSAGES

CODE	MESSAGE	EXPLANATION
00257	INVALID DEVICE SPECIFIED	Incorrectly spelling a device name; correct spelling.
00258	INVALID FILE-ID	File-id passed to F/S intrinsics is incorrect: program error.
00259	INVALID FILE ACCESS	Reading a write only file or writing to read only file.
00260	ACTIVE FILE TABLE FULL	No more files can be opened until another file is closed first.
00261	DUPLICATE DEVICES SPECIFIED	More than one of the same device name in multiple device string specification; delete one.
00262	"SOURCE"" = "DESTINATION"	Trying to read and write to the same device; change one of them.
00263	TOO MANY DEVICES SPECIFIED	Can't have more than one source device; can't have more than one destination device for a compare operation.
00264	CONFLICTING I/O	A write to display from I/O buffer is occurring during a GET operation.

COMPARE COMMAND ERROR MESSAGES

CODE	MESSAGE	EXPLANATION
00265	DIFF. LENGTH RECORDS	Record lengths on source and destination devices are different on a compare operation.
00266	DIFF. IN BYTE x RECORD y FILE z	A mismatch on a byte on a compare operation.
00267	DIFFERENCE IN RECORD TYPE	Record types did not agree from source and destination devices.

COMMAND HANDLER ERROR MESSAGES

CODE	MESSAGE	EXPLANATION
00268	ILLEGAL PARAMETER IN COMMAND	Unrecognized keyword in command parameter list.
00269	EXTRANEOUS PARAMETER IN COMMAND	Too many keywords in command parameter list.
00270	MISSING PARAMETER IN COMMAND	Expected keyword in command parameter list not found.
00271	NON-NUMERIC PARAMETER IN COMMAND	Illegal numeric digit found in command parameter list.
00272	EXCESSIVE NUMBERIC PARAMETER IN COMMAND	Numeric digit too large in command parameter list.

ASSIGN INTRINSIC ERROR MESSAGES

CODE	MESSAGE	EXPLANATION
00273	ASSIGN TABLE FULL	Can't fit new assignment into assign table; delete another entry to make room.
00274	NAME NOT IN ASSIGN TABLE	Name requested wasn't found in assign table.
00275	RE-ASSIGNMENT NOT ALLOWED	Devices cannot be reassigned in Edit Mode or Data Logging Mode. To reassign devices, disable Edit or Data Logging Mode, reassign, then enable the mode.
00276	ILLEGAL ASSIGN NAME	incorrect character used in assign name. Names are truncated to 11 characters.

COMMAND INTERPRETER ERROR MESSAGES

CODE	MESSAGE	EXPLANATION
00277	PROGRAM NOT FOUND	Unable to find program with name specified in command channel.
00278	UNRECOGNIZED COMMAND	Unable to find command to match command specified in command channel.
00279	APPLICATION NOT FOUND	Unable to find a running application program.
00280	EXECUTE FILE NOT FOUND	Unable to find an active execute file.
00281	VOLUME TABLE FULL	Too many devices currently active. De-activate one of the devices.

EXTERNAL PRINTER ERROR MESSAGES

CODE	MESSAGE	EXPLANATION
00336	PRINT FAIL	General external printer fail; unspecific.
00337	NO PAPER ON EXTERNAL PRINTER	Paper out; resupply paper.
00338	NO EXTERNAL PRINTER	No printer connected, or no PCA board.

DATACOMM ERROR MESSAGES

CODE	MESSAGE	EXPLANATION
00352	TERMINAL NOT IN REMOTE	Terminal remote switch not down.
00353	DATACOMM ERROR	Unspecific datacomm transmit or receive error.

SHARED PRINTER ERROR MESSAGES

CODE	MESSAGE	EXPLANATION
00368	NO SHARED PRINTER	No shared printer connected to HP-IB.
00369	PRINTER IS BUSY, RETRY COMMAND WHEN FREE	Shared printer is currently being used; wait until free.
00370	NO READ FROM PRINTER	Can't read from shared printer.
00371	NO PP	Printer did not respond in time.

HP-IB ERROR MESSAGES

CODE	MESSAGE	EXPLANATION
00384	HP-IB TIME-OUT	Data transfer didn't complete before allotted time.
00385	NO HP-IB PCA	No HP-IB PCA in terminal.
00386	ILLEGAL HP-IB ADDR	HP-IB address specified did not exist.
00397	HP-IB DEV BUSY, RETRY COMMAND WHEN FREE	Shared HP-IB device currently busy; wait until free.
00388	NO HP-IB CONTROLLER	No terminal on the HP-IB has responded to a request for control.
00389	HP-IB TEST FAIL xx	Self test of HP-IB failed.
00390	HP-IB TEST NOT ATTEMPTED, ADDR=x, SYSCTL=YES or NO, CIC=YES or NO	Self-test of HP-IB cannot be performed under current conditions or configurations.
00392	NOT HP-IB SYSTEM CONTROLLER	HP-IB protocol violated; REN and IFC control lines may only be accessed by system controller.

TERMINAL ERROR MESSAGES

CODE	MESSAGE	EXPLANATION
00400	HP-IB TE#x TO TE#y TEST FAIL	Transfer from terminal to terminal failed over HP-IB.

DISC ERROR MESSAGES

CODE	MESSAGE	EXPLANATION
449	File name not found	File name does not exist on the volume, or it exists as a purged file. Use SHow Files to list the unpurged files on the LOG device; use SHow Purged or SHow Purged <filename> to list the purged files(s) on the LOG device. If the file is contained on the volume mounted on disc drive #2, the volume name must be specified also in the command.
450	File name already exists	Attempted to Copy, Transfer, or Record data to a file already present on the volume. Either specify a new filename in the command, REName the existing file, or Purge the existing file and execute the command.
451	Illegal file name	Attempted to use illegal characters in the file name; they must be either alpha, numeric, underscore (_), or apostrophe(').
452	File size too large	Attempted to CReate a file that is too large for the remaining space on the volume. Either PAck the diskette if there are any purged files, use a new volume, or reduce the file size.
453	Volume name not found	Use SHow Volumes to show correct volume name(s) in drives(s).
454	Volume name already exists	Attempted to name a volume with the same name as the volume on the other drive.
455	Illegal volume name	Attempted to use illegal characters in the volume name when using the CReate or RENname commands. They must be either alpha or numeric.
456	Volume name not allowed	Attempted to specify a volume name or disc number as part of the new file name when executing the RENname File command. Omit the volume name when specifying the new file name.
457	Volume is write-protected	The write-enable notch on the flexible disc is covered. Remove the tab covering the notch.
458	Volume has changed	The volume was replaced with another volume while a file was open (i.e., while a file-read or a file-write was suspended).
459	Directory is full	Attempted to add more than 240 files to a volume. If the volume contains purged files, PAck the diskette to gain file entry space.
460	Improper directory data	The volume has been removed from the drive with a file open. The file that was left open is now being accessed.
461	Disc PACK not allowed	A file is open (i.e., PACK command was attempted while a file-read or a file-write was suspended). Probably a CTU emulator file is still active; execute the REwind command, or exit Edit Mode, then execute the PAck command.
462	CREATE VOLUME not allowed	Volume has not been purged: purge the volume, then execute the CReate Volume command.
463	FORMAT VOLUME not allowed	Attempted to format an unpurged volume; purge the volume, then execute the FORmat Volume command.
464	RENAME FILE not allowed	The file is open. (i.e., RENname File command was attempted while a file-read or a file-write was suspended. Probably, a CTU emulator file is still active; execute the REwind command, or exit Edit Mode, then execute the RENname File command.)
465	RENAME VOLUME not allowed	A file is open. (i.e., RENname Volume command was attempted while a file-read or a file-write was suspended. Probably, a CTU emulator file is still active; execute the REwind command, or exit Edit Mode, then execute the RENname Volume command.)

DISC ERROR MESSAGES (Continued)

CODE	MESSAGE	EXPLANATION
466	PURGE FILE not allowed	The specified file is open. (i.e., Purge file command was attempted while a file-read or a file-write was suspended. Probably, a CTU emulator file is still active; execute the Rewind command, or exit Edit Mode, then execute the Purge File command.)
467	PURGE VOLUME not allowed	A file is open. (i.e., Purge Volume command was attempted while a file-read or a file-write was suspended. Probably, a CTU emulator file is still active; execute the Rewind command, or exit Edit Mode, then execute the command.)
468	End of disc file	Not enough space left in the file for data during an APpend. Copy the file data to a new file, then execute the APpend command.
469	Disc data did not verify	The Enable Verify command (write-backspace-read mode) is active and the data did not write properly after four retries on the diskette. The diskette may be faulty; insert a new, formatted diskette to continue. If problem continues, call your local Hewlett-Packard service representative.
470	Disc data did not compare	Data bytes are not the same on the two volumes while executing the COmpare Volume command. Possible faulty diskette. Recopy volume data to a new diskette.
471	Bad disc unit	"DISC#" was not used or misspelled, or no disc number given, in the command sequence.
472	Illegal unit number	A character other than "1" or "2" ws used with "DISC#".
473	Disc SEEK FAIL	Door may be closed on drive; close door. Diskette may be faulty or hardware failure. Use new diskette; if problem continues, call your local Hewlett-Packard service representative.
474	Disc WRITE FAIL	Diskette may be faulty or hardware failure. Use new diskette; if problem continues, call your local Hewlett-Packard service representative.
475	Disc READ FAIL	Diskette may be faulty or hardware failure. Use new diskette; if problem continues, call your local Hewlett-Packard service representative.
476	Disc FORMAT FAIL	Diskette may be faulty or hardware failure. Use new diskette; if problem continues, call your local Hewlett-Packard service representative.
477	Disc VERIFY FAIL	The track address data or the track CRC data on the volume did not verify while executing the VERify Volume command. Possible faulty diskette. Data on the volume can be recovered only by a qualified Hewlett-Packard service representative using the DIAGNOSTIC commands.
478	Disc INITIALIZE FAIL	The disc controller failed to mark a bad track. Probable hardware failure; call your local Hewlett-Packard service representative.
479	Disc controller time-out	Disc drive not connected to terminal, or controller PCA not present in terminal. Check cable connections; check presence of controller in terminal.
480	Bad disc DSJ	Possible hardware failure; call your local Hewlett-Packard service representative.
481	Illegal disc address	Diskette may be faulty or hardware failure. Use new formatted diskette, if problem continues, call your local Hewlett-Packard service representative.
482	No Disc controller attached	Disc controller PCA not present in terminal.

DISC ERROR MESSAGES (Continued)

CODE	MESSAGE	EXPLANATION
483	Disc self test FAIL, Unit is <u>, Test(Subtest) is <t> (<s>), Head is <h>	Refer to "Self Test" section.
484	Disc self test FAIL, data buffer compare error	Refer to the "Self Test" section.
485	Disc control block not available	These messages provide the Hewlett-Packard service representative diagnostic information when executing the DIAGNOSTIC commands.
486	Disc control block not found	
488	Bad buffer pointer	
489	Illegal date	Attempted incorrect format for date; use <mm>/<dd>/<yy> in SET Date command.
490	DIAL not allowed	Attempted to execute DIAL command with the modem not connected, the incorrect datacomm PCA installed, or the multipoint ROMs installed.
491	Phone connection not established	Autodial wa unsuccessful (i.e., computer modem did not answer). Phone line may be busy, or computer may be down.
492	ILLEGAL INTRINSIC CALL	(Diagnostic message for the Hewlett-Packard service representative.)
495	COPY VOLUME not allowed	The "to" volume has not been purged. Purge the volume to remove the volume name, then execute the Copy Volume command.
496	File types clash	You tried to compare an ASCII file with a binary file. (Files must be of the same type.)
497	No disc in drive	Attempted command needed a volume for its execution. Put volume in drive.
498	No more disc space	The volume is full. If the volume contains purged files, PAck the volume to make more space available; otherwise, replace it with another volume that has more space.
499	File is already open	Attempted to open an already open file.
500	Bad disc format	Any on-line volume that could satisfy a request is either unformatted or unlabeled.
501	Unexpected end of file	Incorrectly formatted disc file. (Disc may have been removed from drive while file was being written.)

TERMINAL SPECIFICATIONS

APPENDIX B

INTRODUCTION

This appendix contains the terminal specifications, including general specifications, data comm, power requirements, physical measurements, environmental conditions, and product regulations.

Specifications

GENERAL

Screen Size:	150 mm (6 inches) × 215 mm (8.5 inches)
Screen Capacity:	24 lines × 80 columns [A], 360 dots × 720 dots [G]
Character Generation:	7 × 9 enhanced dot matrix, 9 × 15 dot character cell, non-interfaced raster scan
Character Set:	128 characters upper/lower case, control codes, Math (integral sign, Greek letters, etc.). Line drawing, Large characters
Cursor:	Blinking-underline [A], Crosshair [G]
Display Enhancements:	White on black, black on white, half-bright, underline, and blinking
Refresh Rate:	60 Hz or 50 Hz
Tube Phosphor:	P4
Implosion Protection:	Bonded implosion panel
Memory:	115 lines of 80 characters max. [A], 360 dots × 720 dots of displayable points [G], 64K bytes BASIC workspace
Keyboard:	Detachable, 8 user definable softkeys, 18 control and editing keys, graphics pad, cursor pad, auto-repeat, n-key rollover, 1.2m (4 foot) cable
Option Slots:	6 available
5¼" Flexible Disc Drive System:	:256 data bytes/sector, 16 sectors/track, 35 tracks/surface :2 surfaces/disc, 270K bytes/disc :Normal disc media life: 200 hours :Data transfer speed 1K bytes/sec :HP Logical Interchange Format :Up to 2 drives supported on 2647F NOTE: Cartridge tapes are not supported.
	[A] Alphanumeric, [G] Graphics

DATA COMMUNICATIONS

Data Rate:	110, 150, 300, 1200, 2400, 4800, 9600 baud external switch selectable. Operations above 1200 baud may require nulls or handshake protocol to ensure data integrity. External clocking requires a TTL signal 16 × bps.
Vector Drawing Time:	9600 baud (typical), 7 ms half screen, 10 ms full screen
Communication Interface:	:Asynchronous, point-to-point RS232C; compatible with Bell 103A modems; limited compatibility with Bell 202D/S/T modems. Choose either main or reverse channel line timearound for half-duplex operation; CCITT V.24. XON/XOFF handshake protocols supported in addition to standard HP handshake protocols. Current loop, split speed, custom baud rates (optional). NOTE: Multipoint communications is not supported.
Transmission Modes:	Full- or half-duplex, asynchronous
Operating Modes:	On-line, off-line character, block
Parity:	Even, odd, none; switch selectable
Peripheral Support:	The shared peripheral interface allows the user to connect up to 8 of the following qualified peripheral devices only: 2631B/G, 2671A/G, 2673A, 7225B, 7245B, 7470A, 7580A, 9872 C,T, 9876A, and 2647F terminals. The maximum overall interconnecting cable allowed is 15m. RS232C serial printer interface (optional) HP 8-bit duplex parallel printer interface (optional) Video outport interface (optional)

POWER REQUIREMENTS

Input Voltage:	115V (+10%, −23%) at 50/60 Hz (±0.2%) 230V (+10%, −15%) at 50 Hz (±0.2%)
Power Consumption:	140W to 180W

PHYSICAL SPECIFICATIONS

Keyboard Weight:	32 kg (7 lbs.)
Keyboard Dimensions:	444 mm (W) × 216 mm (D) × 90 mm (H), (17.5 inches (W) × 8.5 inches (D) × 3.5 inches (H))

ENVIRONMENTAL CONDITIONS

Temperature (Free Space Ambient):	−10 degrees to 50 degrees C (14 degrees F to 122 degrees F); non-operating 5 degrees to 40 degrees C (41 degrees F to 104 degrees F); operating
Humidity:	20% to 80%; non-condensing
Altitude:	Sea level to 7620m (25,000 ft.); non-operating Sea level 4572m (15,000 ft.); operating
Vibration:*	0.38 mm (0.015 inch) pp, 10 Hz to 55 Hz, 3 axis; non-operating
Shock:*	30 g, 11 ms ½ sine; non-operating
	*Type tested to qualify for normal shipping and handling in original shipping container.

PRODUCT REGULATIONS

Product Meets:	United States: UL requirements for EDP equipment, office appliances and teaching machines. FCC (RFI). Canada: CSA requirements for EDP equipment. Switzerland: SEV Germany: FTZ (RFI) Finland: FTI IEC 380

OPTIONS AND ACCESSORIES

APPENDIX C

INTRODUCTION

This appendix contains information on options and accessories available for the terminal.

PRODUCT NUMBER	DESCRIPTION
2647F	INTELLIGENT GRAPHICS WORKSTATION 720×360 dot graphics image memory and random access alphanumeric memory (12K); BASIC interpreter with graphic commands (64K program workspace); automatic plotting; 128-character Roman, line drawing, math, and large character sets; half-bright, inverse video, blinking, and underline display enhancements; 8 user definable softkeys; RS232 communications interface*; shared peripheral interface; integrated flexible mini disc drive; 4 option slots. Includes one HP-IB cable.
	*NOTE: Interface cable to system not included.
#015	50 Hz, 230V
#016	50 Hz, 115V
#032	EXTENDED ASYNCHRONOUS COMMUNICATIONS Upgrade which provides either RS232C or 20mA current loop communications for the 2647F. Has split speed and custom baud rates.
#035	DIRECT CONNECT TYPE 422 COMMUNICATIONS Converts electrical signals from RS232 to RS422.
#072	SECOND FLEXIBLE MINI DISC DRIVE Adds second non-integrated disc drive unit. (Includes cable for connection to first drive.)

ACCESSORIES	DESCRIPTION
13238A	TERMINAL DUPLEX REGISTER Adds support for HP 8-bit duplex parallel printers. Does not include interface cable.
13254A	VIDEO OUTPUT INTERFACE Generates a video output signal which can be input to a compatible television monitor or video hardcopy unit.
13257G	GRAPHICS PRESENTATION PAC/47 Enhanced Autoplot/47, flowchart generation, and electronic symbol generation.
13257H	PROJECT MANAGEMENT PAC/47 Critical path analysis, time-events chart, and financial decision programs.
13257L	STATISTICAL ANALYSIS PAC/47 Solutions to frequently used statistical operations.
13257M	MATHEMATICAL ANALYSIS PAC/47 Solutions to frequently used mathematical operations.
13257N	LINK/47 Transfers ASCII and binary files between HP 2647F and HP 3000.
13257P	2647F FILE CONVERSION KIT Converts HP 2647A cartridge tape files to HP 2647F disc file format. (Includes 13232U cables. 2 datacomm cables required but not included.)
13257W	WORD/47 Word Processing, page formatting, and forms design software.
13265A #001	300 BAUD MODEM Full duplex, Bell 103/113 compatible, originate only modem. Dual connect and local loopback test. Includes cable to connect terminal to modem.
92190A	FLEXIBLE MINI DISCS (Order from HP Computer Supplies Operation.) Box of 10 mini discs.
CABLES	
13232C	RS232C Connector (Female, 5 Ft.)
13232F	Current Loop Connector Kit (5 Ft.)
13232G	13250B to RS232 Printer (Male, 15 Ft.)
13232H	13250B to RS232 Printer (Female, 15 Ft.)
13232I	Direct Connect RS422 (Male, 16 Ft.)
13232K	13254 to Video Hardcopy Printer (25 Ft.)
13232L	13254 to Compatible TV monitor (25 Ft.)
13232M	13250B to European Modem
13232N	103/202/201/208/209 Modem (Male, 15 Ft.)
13232X	Direct Connect RS232 (Male, 16 Ft.)

ESCAPE CODES

APPENDIX D

INTRODUCTION

This appendix contains information on escape codes recognized by the terminal.

Table D-1. Escape Codes

KEY	CODE	FUNCTION
DISPLAY CONTROL		
↑	ESC A	Cursor up
↓	ESC B	Cursor down
→	ESC C	Cursor right
←	ESC D	Cursor left
BACK SPACE	BS (H^C)	Cursor left one space
CNTL ↖	ESC F	Cursor home down
— — —	ESC G	Cursor return
↖	ESC h	Cursor home (excluding transmit-only fields)
— — —	ESC H	Home cursor (including transmit-only fields)
RETURN	CR (M^C)	Move cursor to left margin
— — —	LF (J^C)	Move cursor down one line
TAB	HT (I^C) ESC I	Forward cursor to next tab position
CNTL TAB or CNTL BACK SPACE	ESC i	Back tab
SET TAB	ESC 1	Set tab at the current cursor column
CLEAR TAB	ESC 2	Clear the tab at the current cursor column
CNTL CLEAR TAB	ESC 3	Clear all tabs
CNTL ←	ESC 4	Set left margin
CNTL →	ESC 5	Set right margin
CNTL ↑	— — —	Set top margin (Forms Design Mode)
CNTL ↓	— — —	Set bottom margin (Forms Design Mode)
CLEAR DSPLY	ESC J	Clear memory from cursor position to end of memory
CNTL CLEAR DSPLY	ESC K	Clear line from the cursor to end of line
ROLL UP	ESC S	Scroll the display up one line
ROLL DOWN	ESC T	Scroll the display down one line
NEXT PAGE	ESC U	Display the next 24 lines of memory
PREV PAGE	ESC V	Display the previous 24 lines of memory
CNTL PREV PAGE	ESC , cDI W# <n>	Display the next display-workspace
CNTL f1	ESC & d	Turn on display enhance (see page A-3)
CNTL f2	ESC [	Start an unprotected field
CNTL f3	ESC]	End an unprotected field or transmit-only field
CNTL f4	ESC W (on)	Turn format mode on. Only unprotected fields can be modified.
CNTL f5	ESC X (off)	
CNTL f6	ESC {	Start transmit-only field
— — —	ESC 6	Alphabetic only field
— — —	ESC 7	Numeric only field
— — —	ESC 8	Alphanumeric field
EDITING GROUP		
INSERT LINE	ESC L	Insert a blank line
DELETE LINE	ESC M	Delete line containing cursor
DELETE CHAR	ESC P	Delete character at cursor
CNTL DELETE CHAR	ESC O	Delete character with wraparound from next line
INSERT CHAR & indicator	ESC Q (on) ESC R (off)	Insert succeeding inputs at cursor
CNTL INSERT CHAR	ESC N (on) ESC R (off)	Character Wraparound Mode. Insert succeeding inputs at cursor with wraparound to next line.
TERMINAL CONTROL GROUP		
ESC	ESC [	Leads off an ASCII escape sequence
CNTL	— — —	Used to generate ASCII control codes and alternate key functions
CAPS LOCK	ESC&k0C (off) ESC&k1C (on)	Upper-case alphabetical lock
MEMORY LOCK & indicator	ESC l (on) ESC m (off)	Memory overflow protect; display lock
AUTO LF	ESC&k0A (off) ESC&k1A (on)	Line Feed with each terminal carriage return
REMOTE	ESC&k0R (off) ESC&k1R (on)	Remote (on-line) operations; otherwise, off-line operation
BLOCK MODE	ESC&k0B (off) ESC&k1B (on)	Block Mode: data displayed but not transmitted until requested; otherwise, terminal is in Character Mode and data transmitted as typed
ENTER	— — —	Enables block transfers
BREAK	— — —	Transmits BREAK signal to interrupt computer
TRANSMIT indicator	— — —	Data link exists
DISPLAY FUNCTIONS & indicator	ESC Y (on) ESC Z (off)	Control functions disabled and displayed
CNTL DISPLAY FUNCTIONS	ESC y (on) ESC Z (off)	Monitor Mode: display all codes received from data comm lines
RESET TERMINAL	ESC 9	(First press): frees the keyboard and clears I/O operations
RESET TERMINAL	ESC E	(Second press): sets the terminal to power-on state
TEST	ESC z	Terminal Self-Test (no disc test)

NOTE: Unless otherwise specified, all display control functions apply to the present display-workspace only (i.e., the other three display-workspaces are not affected.)

Table D-1. Escape Codes (Continued)

KEY	CODE	FUNCTION
	ADDITIONAL FUNCTIONS	
— — —	ESC E (E^C)	Enquiry from the computer
— — —	ESC F (F^C)	Acknowledge — response to ENQ
— — —	ESC G (G^C)	Bell
— — —	ESC) <x>	Selects the active character set: x — Character Set @ — Standard character set A, B, or C — Math, Line Drawing, or Large Character Set, depending on the setting of switches C1, C2, and C3 on the Display Timing/Control PCA (refer to Configuration section of the Reference Manual).
— — —	ESC N (N^C)	Turn on alternate character set
— — —	ESC O (O^C)	Turn off alternate character set
— — —	ESC Q (Q^C)	Block transfer trigger
— — —	ESC R (R^C)	Block transfer enable from terminal
— — —	ESC ^ (^^C)	Record separator
— — —	ESC _ (_^C)	Unit separator
— — —	ESC & 0	Delay one second
— — —	ESC '	Cursor sensing (screen relative)
— — —	ESC a	Cursor sensing (absolute)
— — —	ESC b	Keyboard enable
— — —	ESC c	Keyboard disable
— — —	ESC d	Block transfer enable from computer (See DC2)
— — —	ESC e	Fast binary read
— — —	ESC f	Modem disconnect
[ ] [NEXT PAGE]	ESC] (on) ESC k (off)	Display user-defined soft key menu
— — —	ESC ^	Terminal status
— — —	ESC ~	Extended status request
	TERMINAL CONTROL	
— — —	ESC & a <parameters>	Cursor addressing
Example:	Cursor to 12th row (absolute) 35th column (+, - for relative addressing) ESC & a 12r 35C	
Example:	Cursor to 16th row (screen) 70th column (+, - for relative addressing) ESC & a 16y 70C	
— — —	ESC & b <parameters> or ESC & c <parameters>	HP diagnostics ONLY

KEY	CODE	FUNCTION
	TERMINAL CONTROL (Cont.)	
[ ] [f1] through < [@] through [O] >	ESC & d <enhancement>	Turn on display enhancement

where: enhancements = @ through O

Example: Select half-bright, blinking, and underline.
ESC & d M

	Enhancement Character													
	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Half Bright									x	x	x	x	x	x	x	x
Underline					x	x	x	x					x	x	x	x
Inverse Video			x	x			x	x			x	x			x	x
Blinking		x		x		x		x		x		x		x		x
End Enhancement	x															

— — — ESC & f <parameters> — Define soft keys

where:

<parameters> = {0-8}k (0 (normal), 1 (local only), 2 (transmit only)) a {1-8}d {1-80}L <label string> <text string>

Example: Assign the string "HELLO-MYACNT↵" to the [f1] key. The key should function as normal keyboard input.

ESC & f 1 k 2 a 6d 13 L LOG-ON HELLO-MYACNT↵

— — — ESC & f <key #>E — Execute the soft key. Key # = 0-8

KEY	DEFAULT	PROGRAMMABLE
[RETURN]	↵	
[f1]	ESC p to computer	
[f2]	ESC q to computer	
[f3]	ESC r to computer	
[f4]	ESC s to computer	Up to 80-character sequence for each key (local, transmit or both)
[f5]	ESC t to computer	
[f6]	ESC u to computer	
[f7]	ESC v to computer	
[f8]	ESC w to computer	

Table D-1. Escape Codes (Continued)

KEY	CODE	FUNCTION
	TERMINAL CONTROL (Cont.)	
———	ESC & g <parameters ··	Simulate PA, PF keys
———	ESC & k <parameters ··	Define latching key settings.

where:

<parameter ·· = 0 (up) { a (Auto LF), b (Block Mode), c (Caps Lock), r (Remote) } 1 (down)

Example: Block Mode up Remote up Auto LF down Caps Lock down

ESC & k 1 a 0 b 1 c 0 R

[redacted]	———	In REMOTE, transfers data from source device to computer. In LOCAL, transfers one file from source device to DISPLAY.
[redacted]	———	In REMOTE, transfers data from computer to destination device. In LOCAL, transfers one file from DISPLAY to destination device.
[redacted]	———	In REMOTE, enables block transfers. In LOCAL, operates same as [redacted].
———	ESC & s <parameters ··	Define Processor PCA strap settings. (A–Z)

Example: Inhibit the self test function (L switch open) and set the Line/Page Switch to Page (D switch open).

ESC & s 0 1 0 D

[redacted]	ESC & j B (on) ESC & j @ (off)	Turn on Softkey Label Line. Turn off Softkey Label Line.
———	ESC & p <code> p21C	Define end-of-data character.

<code> = 1 thru 127: The corresponding ASCII code character is defined as the end-of-data character when it appears at the start of a record in a Record, "Copy File from DAtacomm," or "Copy All from DAtacomm" operation.

<code> = 0 or none: The end-of-data feature is disabled.

Note: This sequence always returns an "S" (successful) indication.

———	ESC & p <code> p20C	Initiate Record mode and define end-of-data character.

<code> = 1 thru 127: Record mode is initiated and the ASCII code character corresponding to "code" is defined as the end-of-data character. When Record mode is ended, the previous end-of-data character becomes active.

= 0: The end-of-data feature is disabled for this operation only.

= none: The currently selected end-of-data character (if any) is used.

Note: 1. This sequence returns an "S" indication only if the Record operation was successful.
2. A DC1 (completion code) should be transmitted following the end-of-data character.

KEY	CODE	FUNCTION
	TERMINAL STATUS	
———	ESC ^	Primary Terminal Status.

Example: Request primary terminal status.

Program sends: ESC ^

Terminal responds: ESC \ <7 status bytes ··

where:

byte 0 = Display size
byte 1 = Processor PCA Switches (A–D)
byte 2 = Processor PCA Switches (E–H)
byte 3 = Latching Keys
byte 4 = Transfer Pending Flags
byte 5 = Error Flags
byte 6 = Device Transfer Pending Flags

———	ESC ~	Secondary Terminal Status

Example: Request secondary terminal status.

Program sends: ESC ~

Terminal responds: ESC ! <7 status bytes ··

where:

byte 0 = Buffer Memory
byte 1 = Terminal Firmware Configuration
byte 2 = Processor PCA Switches (J–M)
byte 3 = Processor PCA Switches (N–R)
byte 4 = Processor PCA Switches (S–V)
byte 5 = Processor PCA Switches (W–Z)
byte 6 = Memory Lock Bilingual Mode

Note: Terminal must receive a DC1 character to transmit status bytes if G and H switches on the Processor PCA are closed.

———	ESC & p <device code ··^	Device Status

where:

<device code ·· =
1 = left tape
2 = right tape
4 = external printer

Terminal responds:

ESC & p <device code ·· <3 status bytes ··

Example: Request device status of left tape.

Program sends: ESC & p 1 ^

Terminal responds: ESC & p 1 <3 status bytes ··

———	ESC * s 1 ^	Read id (response is 2647F <terminator ··).

Example: Read device id.

Program sends: ESC * s 1 ^ D_1

Terminal responds: 2647F <terminator ··

———	ESC & b <bank id ··p^ or ESC & c <bank id ··p^	Read memory size on specified memory bank.

Example: Read memory size on bank 1.

Program sends: ESC & b 1 p ^

Terminal responds: <no. bytes of memory ·· <terminator ··

Note: The default bank id is 2.

Table D-1. Escape Codes (Continued)

KEY	CODE	FUNCTION

GRAPHICS CONTROL SEQUENCES

ESC • <control sequence>

- b = Raster dump
- d = Display control
- l = Graphics text label
- m = Mode control
- p = Plot control
- r = Raster dump
- s = Status
- t = Compatibility mode

DISPLAY CONTROL

ESC • d <parameters>

KEY	CODE	FUNCTION
SHIFT CLEAR	a	Clear graphics memory
— — —	b	Set graphics memory
SHIFT G DSP	c	Turn on graphics display
SHIFT G DSP	d	Turn off graphics display
SHIFT A DSP	e	Turn on alphanumeric display
SHIFT A DSP	f	Turn off alphanumeric display
ZOOM	g	Turn on zoom
ZOOM	h	Turn off zoom
ZOOM IN or ZOOM OUT	<size> i	Set zoom size (1-16)
— — —	<x,y> j	Set zoom position
CURSOR	k	Turn on graphics cursor
CURSOR	l	Turn off graphics cursor
SHIFT RB LN	m	Turn on rubber band line
SHIFT RB LN	n	Turn off rubber band line
— — —	<x,y> o	Move graphics cursor absolute
◄ , ▲ , ► , ▼	<x,y> p	Move graphics cursor incremental
— — —	q	Turn on alphanumeric cursor
— — —	r	Turn off alphanumeric cursor
SHIFT TEXT	s	Turn on graphics text mode
RESET	t	Turn off graphics text mode
	z	NOP

Example: Clear the graphics display, position the cursor at x=100, y=100, turn the cursor on, and zoom to 4 times.

```
ESC • d a 100,100o k 4i G
```

GRAPHICS LABEL

ESC • l <text label><CR>, <LF>, <ETX>, or <ESC>

Example: Send the text "X=TIME, Y=TEMP"

```
ESC • l X=TIME, Y=TEMP CR
```

KEY	CODE	FUNCTION

VECTOR DRAWING MODE

ESC * m <parameters>

KEY	CODE	FUNCTION
— — —	<mode> a	Select drawing mode (0-4)*
— — —	<line type> b	Select line type (1-11)**
— — —	<pattern scale> c	Define line pattern (2 bytes)
— — —	<pattern> d	Define area shading pattern (8 bytes)
— — —	<x1,y1,x2,y2> e	Fill area, absolute
— — —	<x1,y1,x2,y2> f	Fill area, relocatable
— — —	<x,y> j	Set relocatable origin
— — —	k	Set relocatable origin to current pen position
— — —	l	Set relocatable origin to graphics cursor position
SHIFT SIZE	<size> m	Set graphics text size (1-8)
SHIFT TANG	<rotation> n	Set graphics text orientation (1-4)
SHIFT TANG	o	Turn on text slant
SHIFT TANG	p	Turn off text slant
— — —	< 0-9 > q	Set graphics text origin
— — —	r	Set graphics defaults
	z	NOP

* 0 (no effect), 1 (clear), 2 (set), 3 (complement), 4 (jam)

** 1 (solid line), 2 (user line pattern), 3 (user area pattern), 4 (line #1), 5 (line #2), 6 (line #3), 7 (line #4), 8 (line #5), 9 (line #6), 10 (line #7), 11 (point plot)

Example: Select the set drawing mode, a graphics text size of 2 and slanted. Set the text to be center justified.

```
ESC • m 1a 2m o 4Q
```

PLOTTING COMMANDS

ESC • p <parameters>

— — —	a	Lift the pen
— — —	b	Lower the pen
— — —	c	Use graphics cursor as new point
— — —	d	Draw a point at the current pen position and lift the pen
— — —	e	Set relocatable origin to the current pen position
— — —	f	Data is ASCII absolute
— — —	g	Data is ASCII incremental
— — —	h	Data is ASCII relocatable
— — —	i	Data is absolute
— — —	j	Data is short incremental
— — —	k	Data is incremental
— — —	l	Data is relocatable
— — —	z	NOP

Example: Draw a box 25 units wide and 10 units high, beginning at x=100, y=50.

```
ESC • p a f 100 50 g 25,0 0,10 -25,0 0,-10Z
```

Table D-1. Escape Codes (Continued)

KEY	CODE	FUNCTION
	RASTER DUMP	
	ESC * r <parameter>	
— — —	a	Start transfer
— — —	b	End transfer
— — —	c	Erase screen
— — —	d	Turn on video
— — —	e	Return raster status
— — —	i	Set all parameters to default values
— — —	j	Return raster size status ("720,360")
— — —	k	Return model number ("2647F")
— — —	< no. of dots> m	Window horizontal address
— — —	< no. of dots > n	Window vertical address
— — —	< no. of dots> p	Window horizontal dimension
— — —	< no. of dots> q	Window vertical dimension
— — —	< no. of dots> s	Horizontal size of picture
— — —	< no. of dots> t	Vertical size of picture
— — —	< no. of dots> x	X offset
— — —	< no. of dots> y	Yofffset
— — —	z	NOP
	ESC * b <parameter>	
— — —	<no. of data bytes> r	Read specified number of data bytes (scan line) to datacomm.
— — —	<no. of data bytes>- w	Write specified number of bytes into graphics memory scan line.
	GRAPHICS STATUS	
	ESC * s <parameter> ^	
— — —	1	Read device I.D.
— — —	2	Read pen position
— — —	3	Read graphics cursor position
— — —	4	Read cursor position and wait for key
— — —	5	Read display size
— — —	6	Read graphics capabilities
— — —	7	Read graphics text status
— — —	8	Read zoom status
— — —	9	Read relocatable origin
— — —	10	Read reset status
— — —	11	Read area shading
— — —	12	Read dynamics

Example: Read text status.
Esc * s 7 ^ DC1

KEY	CODE	FUNCTION
	COMPATIBILITY MODE	
	ESC * t <parameter>	
— — —	<0/1/2> a	Set graphics input terminator (0=CR, 1=CR EOT, 2=none)
— — —	<0/1> b	Set Page Full Break strap (0=out, 1=in)
— — —	<0/1> c	Set Page Full Busy strap 0=out, 1=in)
	z	NOP

Processor PCA switches
P open = Scaled compatibility mode.
Q open = Unscaled compatibility mode.

Example: Select a CR input terminator and set the Page Full Busy strap.
Esc * t 0a 1C

Table D-1. Escape Codes (Continued)

COMMAND SYNTAX

NOTE 1: Uppercase characters in the command syntax are acceptable abbreviations (see "Abbreviated Command Sequence" in the center column of the table). Characters in brackets "[ ]" are optional.

NOTE 2: ESC&p escape sequences listed here are compatible with HP 2641A, HP 2645A, HP 2647A, and HP 2648A terminals; however, it is recommended that ESC,c command sequences be used.

NOTE 3: When using the ESC&p format, the <x> and <y> values refer to source device and destination device numbers:

1 = Left tape
2 = Right tape
3 = Display
4 = Printer

LEGEND

<device> defined as:
Left tape
Right tape
EXternal printer
SHared printer#<n> (where 0≤n≤7, i.e., SH#5)
DIsplay
TErminal#<p>
Hp-ib#<p>[#<s>[#<m>]]
where p = primary address
m = module address
s = secondary address
DAtacomm
Null
Graphics

<name> defined as:
Source (only one may be selected)
Destination
LOG
user assigned name for cartridge tapes, printers, and Hp-ib devices)

<volume name> defined as:
user-assigned volume name (up to 6 alpha and/or numeric characters; an alpha character must be first)

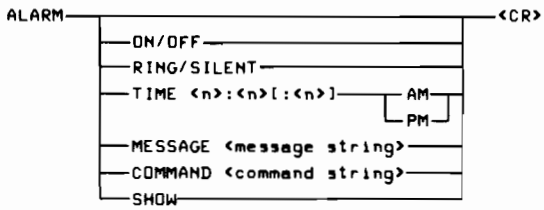
<disc file name> defined as:
user-assigned disc file name (up to 10 alpha and/or numeric characters; an alpha character must be first). The volume name may be specified immediately following the disc file name, separated by a semicolon (;) (no spaces allowed). The disc drive unit number may be specified instead of the volume name. Use either Disc # <n> or D # <n> immediately following the disc file name separated by a colon (:) (no spaces allowed). *The volume name or drive unit number must be included when specifying the names on a volume mounted on disc drive no. 2.*

<disc device> defined as:
DISC#<n>
where n =1 or 2 (the drive unit number)

COMMAND SYNTAX
<commend sequence>

ESC,c **<abbreviated command sequence>** ↵
or
ESC&p **<parameters>**

DESCRIPTION



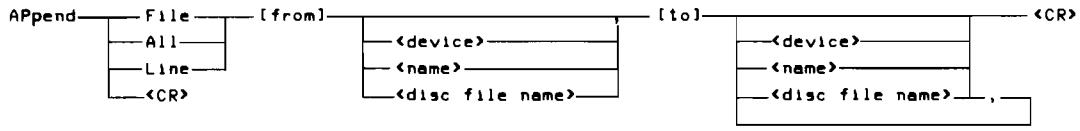
Example: Set the alarm to ring at 10:30 AM and display the message "Meeting in Conference Room 240 — bring progress chart".

```
ALARM TIME 10:30 AM <CR>
ALARM MESSAGE Meeting in Conference Room 240 -
              bring progress chart <CR>
ALARM RING <CR>
ALARM ON <CR>
```

ALARM ↵	ALARM ↵	Displays the status of ALARM ON/OFF and ALARM RING SILENT.
ALARM ON ↵	ALARM ON ↵	Turns on the alarm function.
ALARM OFF ↵	ALARM OFF ↵	Turns off the alarm function.
ALARM RING ↵	ALARM RING ↵	Causes the terminal's bell to ring continuously when alarm time is reached.
ALARM SILENT ↵	ALARM SILENT ↵	Turns off the alarm ring function.
ALARM TIME <n>:<n> [:<n>] AM (or PM) ↵	ALARM TIME <n>:<n> [:<n>] AM (or PM) ↵	Specifies the alarm time. The terminal's internal clock is the reference. (See SET TIme command.)
ALARM MESSAGE <message string> ↵	ALARM MESSAGE <message string> ↵	Allows you to specify a message (up to 31 characters) to be displayed in the message line of the display when alarm time is reached.
ALARM COMMAND <command string> ↵	ALARM COMMAND <command string> ↵	Allows you to specify a command (up to 31 characters) to be executed when alarm time is reached
ALARM SHOW ↵	ALARM SHOW ↵	Displays the current alarm time, alarm message string, and alarm command string.

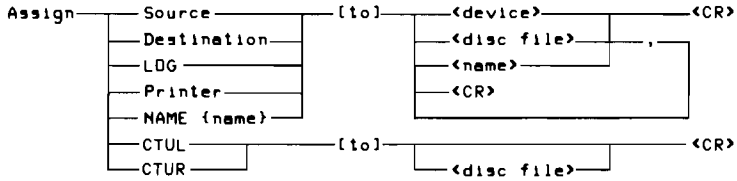
Table D-1. Escape Codes (Continued)

COMMAND SYNTAX <command sequence>	ESC,c<abbreviated command sequence> or ESC&p <parameters>	DESCRIPTION



Example: APpend Display to "CHAPTER 3" disc file.
APpend All Display to CHAPTER 3 ↵

APpend File ↵	ESC,cAP F ↵	Append one file from previously-assigned Source device to previously-assigned destination device.
APpend All ↵	ESC,cAP A ↵	Append all data from previously-assigned Source device to previously-assigned Destination device.
APpend Line ↵	ESC,cAP L ↵	Append a line from previously-assigned Source device to previously-assigned Destination device.
APpend File [from] <device> [to] <device> ↵	ESC,cAP F <device> <device> ↵	Append a file from specified device to specified device. (If the destination is not a disc file, the command performs the same operations as the COPY command.
APpend File [from] <device> [to] <disc file name> ↵	ESC,cAP F <device> <disc file name> ↵	Append a file from specified device to specified disc file.
APpend All [from] <device> [to] <disc file name> ↵	ESC,cAP A <device> <disc file name> ↵	Append all data from specified device to specified disc file.
APpend Line [from] <device> [to] <disc file name> ↵	ESC,cAP L <device> <disc file name> ↵	Append a line of data from specified device to specified disc file.
APpend File [from] <disc file name> [to] <disc file name> ↵	ESC,cAP F <disc file name> <disc file name> ↵	Append a file from specified disc file to specified disc file.
APpend Line [from] <disc file name> [to] <disc file name> ↵	ESC,cAP L <disc file name> <disc file name> ↵	Append a line of data from specified disc file to specified disc file.

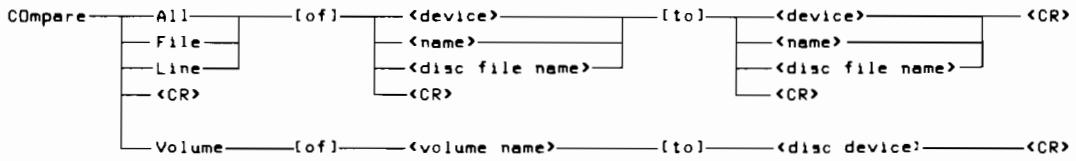


Example: Assign "PLOTTER" to Hp-ib device number 4.
Assign Name PLOTTER to Hp-ib#4 <CR>

Assign Source [to] Left tape ↵	ESC,cA S L ↵ or ESC&p 1S	Assigns left tape as source device.
Assign Source [to] Right tape ↵	ESC,cA S R ↵ or ESC&p 2S	Assigns right tape as Source device
Assign Source [to] Display ↵	ESC,cA S DI ↵ or ESC&p 3S	Assigns display as source device
Assign Source [to] Graphics ↵	ESC,cA S G ↵	Assigns Graphics Memory as source device
Assign Source [to] Hp-ib# p ↵	ESC,cA S H# p ↵	Assigns an HP-IB device as source device
Assign Source [to] disc file name ↵	ESC,cA S disc file name ↵	Assigns specified disc file name as source device.
Assign Destination [to] Left tape ↵	ESC,cA D L ↵ or ESC&p 1D	Assigns left tape as Destination device
Assign Destination [to] Right tape ↵	ESC,cA D R ↵ or ESC&p 2D	Assigns right tape as Destination device
Assign Destination [to] Display ↵	ESC,cA D DI ↵ or ESC&p 3D	Assigns display as Destination device
Assign Destination [to] Graphics ↵	ESC,cA D G ↵	Assigns Graphics Memory as Destination device
Assign Destination [to] EXternal printer ↵	ESC,cA D EX ↵ or ESC&p 4D	Assigns local printer as Destination device
Assign Destination [to] SHared printer#5 ↵	ESC,cA D SH#5↵	Assigns shared printer as Destination device
Assign Destination [to] Hp-ib# n #n #n ↵	ESC,cA D H# n ↵	Assigns an HP-IB device(s) as Destination device(s)

Table D-1. Escape Codes (Continued)

COMMAND SYNTAX <command sequence ·	ESC,c <abbreviated command sequence · ↵ or ESC&p <parameters ·	DESCRIPTION
Assign Destination [to] <disc file name · ↵	ESC,cA D <disc file name > ↵	Assigns specified disc file name as Destination device.
Assign LOG [to] <device · ↵	ESC,cA LOG <device > ↵	Assigns any device as LOG device.
Assign Printer [to] SHared printer# <n · ↵	ESC,cA P SH# <n > ↵	Assigns the shared printer to be associated with the name "PRINTER" of the ESC,c printer escape sequences.
Assign Printer [to] HP-ib# <p · ↵	ESC,cA P H# <p · ↵	Assigns the HP-ib device to be associated with the name "PRINTER" of the ESC,c printer escape sequences.
Assign Printer [to] EXternal Printer ↵	ESC,cA P EX ↵	Assigns the external Printer (connected via the 13238A or 13250B) to be associated with device 4 of the ESC&p 4D printer escape sequences.
Assign CTUL [to] <disc file name · ↵ CTUR	ESC,cA CTUL <disc file name> ↵ CTUR	Assigns a name (up to 6 characters) to be used with tape emulator files on the disc (e.g., CTUL (or CTUR) R002 becomes *ABCDEF*R002.)
Assign Name <user-defined name · [to] <device · ↵	ESC,cA N <name > <device> ↵	Assigns a user-specified name to any device or disc file.
	BYE ——— <CR>	
BYE ↵	ESC,cBYE ↵	Defaults user.group name to "user.group" on shared printer listings.
	CALENDAR ——— [<dd>] ——— <↵> [<mm>/<dd>[/<yy>]]	
CALENDAR ↵	ESC,c CALENDAR ↵	Displays the month set by the SET DATE command.
CALENDAR <dd> ↵	ESC,c CALENDAR <dd> ↵	Highlights the day of the month specified in the command of the month and year specified by the SET DATE command.
CALENDAR <mm>/<dd> ↵	ESC,c CALENDAR <mm>/<dd> ↵	Displays the month and day specified in the command (the day is highlighted) of the year specified by the SET DATE command.
CALENDAR <mm>/<dd>/<yy> ↵	ESC,c CALENDAR <mm>/<dd>/<yy > ↵	Displays the month (and highlights the day) of the year specified in the command.
	CLOse ——— Window#<n> ——— <CR>	
CLOse Window#5 ↵	ESC,cCLO W#5 ↵	Removes the Message Line on the display.
CLOse Window#6 ↵	ESC,cCLO W#6 ↵	Removes the Command Line on the display.
CLOse Window#7 ↵	ESC,cCLO W#7 ↵ or ESC&j0	Removes the Soft Key Label Line on the display.



Example: Compare a file on the display with a file on a device with a user name of "SALES".
COmpare File of Display to SALES <CR>

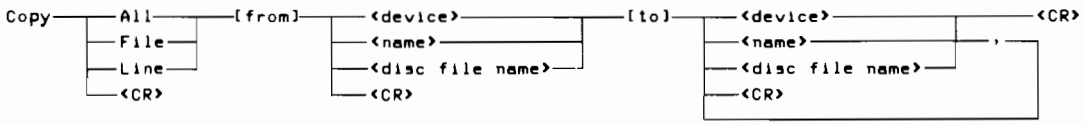
Example: Compare a file on the right tape to disc file name "FORECAST".
COmpare File of Right Tape to FORECAST <CR>

Example: Compare disc volume "ABC" on disc drive no. 2 to disc volume "ABC" on disc drive no. 1.
COmpare Volume of ABC to DISC#1 <CR>

COmpare All [of] <device · [to] <device · ↵	ESC,cCO A <device ·<device · ↵ or ESC&p xs yd 1M	Compare data between devices. (All)
COmpare File [of] <device · [to] <device · ↵	ESC,cCO F <device · <device · ↵ or ESC&p xs yd 1F	Compare data between devices. (File)
COmpare Line [of] <device > [to] <device > ↵	ESC,cCO L <device > <device > ↵ or ESC&p xs yd 1B	Compare data between devices. (Line)

Table D-1. Escape Codes (Continued)

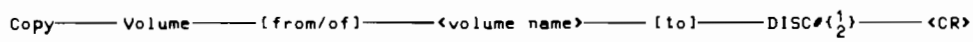
COMMAND SYNTAX <command sequence ·	ESC,c <abbreviated command sequence · ESC or ESC&p <parameters ·	DESCRIPTION
COmpare All ↵	ESC,cCO A ↵ or ESC&p 1M	Compare data between previously-assigned Source and Destination devices. (All)
COmpare File ↵	ESC,cCO F ↵ or ESC&p 1F	Compare data between previously-assigned Source and Destination devices. (File)
COmpare Line ↵	ESC,cCO L ↵ or ESC&p 1B	Compare data between previously-assigned Source and Destination devices. (Line)
COmpare All or File or Line [of]<name> [to]<name> ↵	ESCCO <A or F or L> <name> <name> ↵	Compares data between user-specified device names.
COmpare All or File or Line [of]<disc file name> [to]<disc file name>↵	ESC,cCO <A or F or L> <disc file name> <disc file name> ↵	Compares data between a disc file and a device.
COmpare Volume <volume name> [to] DISC# <n> ↵	ESC,cCO V <volume name> DISC# <n> ↵	Compares data between two disc volumes (requires two disc drives).



Example: Copy the entire contents of the left tape to the shared printer.

```
Copy All from Left tape to SHared printer#5 <CR>
```

NOTE: If you want to copy binary data or graphics data, the TRANSFER command sequence should be used.



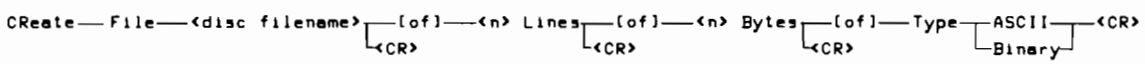
Example: Make a backup copy of a floppy disc with volume name "MEMOS" on another floppy disc which is on disc unit #2. (The new disc will also be named "MEMOS" and cannot be accessed until the original floppy is removed. It will be identified as "duplic" until original floppy disc is removed.)

```
Copy Volume of MEMOS to DISC#2 <CR>
```

Copy File [from] Source [to] DIsplay ↵	ESC,cC F S DI	Transfers one file from Source device to display.
Copy File [from] DIsplay [to] Destination ↵	ESC,cC F DI D	Transfers all data from display workspace to Destination device.
Copy All ↵	ESC,cC A ↵ or ESC&p M	All files (current position) from previously-assigned Source device are transferred to previously-assigned Destination device.
Copy ↵	ESC,cC ↵ or ESC&p xs yd F	One file (current position) from previously-assigned Source device is transferred to previously-assigned Destination device.
Copy Line ↵	ESC,cC L ↵ or ESC&p xs yd B	One line (current position) from previously-assigned Source device is transferred to previously-assigned Destination device.
Copy All [from] <device> [to] <device(s) · ↵	ESC,cC A <device> <device(s) · ↵ or ESC&p xs yd M	All files (current position) from a specified device are transferred to one or more specified device(s).
Copy File [from] <device · [to] <device(s)> ↵	ESC,cC F <device · <device(s) · ↵ or ESC&p xs yd F	One file (current position) from a specified device is transferred to one or more specified device(s).
Copy Line [from] <device · [to] <device(s) · ↵	ESC,cC L <device · <device(s) · ↵ or ESC&p xs yd B	One line (current position) from a specified device is transferred to one or more specified device(s).
Copy All [from] <name · [to] <name · ↵	ESC,cC A <name · <name · ↵	All files (current position) from user-assigned device are transferred to user-assigned device. One file is transferred if disc file.
Copy File [from] <name · [to] <name · ↵	ESC,cC F <name · <name · ↵	One file (current position) from user-assigned device or disc file name is transferred to user-assigned device or disc file name(s).
Copy Line [from] <name · [to] <name · ↵	ESC,cC L <name · <name · ↵	One line (current position) from user-assigned device or disc file name is transferred to user-assigned device or disc file name(s).
Copy Volume [from] <volume name · [to] <disc device · ↵	ESC,cC V <volume name · <disc device · ↵	Copy entire contents of a floppy disc (specified by volume name) to another floppy disc contained on another disc unit (specified by disc device number 1 or 2).

Table D-1. Escape Codes (Continued)

COMMAND SYNTAX <command sequence>	<esc>,c <abbreviated command sequence> or <esc>&p <parameters>	DESCRIPTION



Example: Create a file with the name "DRAFT" on a floppy disc.

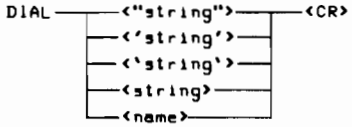
```
CReate file DRAFT
```

CReate File <disc filename> [of] <n> Lines [of] <n> Bytes <esc>	<esc>,cCR F <disc filename> <n> L <n> B <esc>	Creates a file with a user specified file name and a specified number of Lines of a specified number of bytes. (Defaults are 4 lines, 256 bytes per Line.)

Example: Create a volume with the name "FLYER".

```
CReate Volume FLYER on DISC#1 <CR>
```

CReate Volume <volume name> [on] DISC# <n> <esc>	<esc>,cCR V <volume name> DISC# <n> <esc>	Formats the floppy disc (if unformatted), assigns the specified volume name, and clears the directory sectors. (Volume must not be named — see Purge Volume.)



Example: Dial (408) 735-1550.

```
DIAL "(408) 735-1550 <CR>
```

Example: Dial "COMPUTER" (which has its telephone assigned by the ASSIGN command).

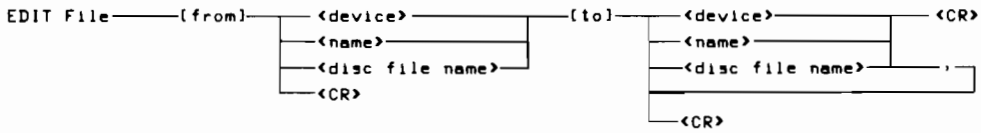
```
DIAL COMPUTER <CR>
```

DIAL <string> <esc>	<esc>,cDIAL <string> <esc>	Causes the HP 13265A Modem Pad to dial the requested number.

Example: Display the message "Insert the tape into the left slot."

```
DIsplay Window#5<CR>
Insert the tape into the left slot <CR>
```

DIsplay Window# <n> <esc>	<esc>,cDI W# <n> <esc>	Display workspace 1, 2, 3, or 4, Message Line (5), Command Line (6), or Soft Key Label Line (7).
	<esc>&jB	Display Soft Key Label Line (7).



Example: Edit disc file "REPORT1" to disc file "REPORT2".

```
EDIT File from REPORT1 to REPORT2 <CR>
```

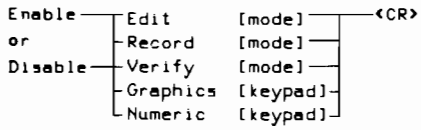
Example: Type draft copy on the display and have it roll off to disc file "MEMO".

```
EDIT File from DIsplay to MEMO <CR>
```

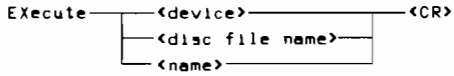
EDIT File [from] <device> [to] <device> <esc>	<esc>,cEDIT F <device> <device> <esc>	Allows user editing of file from specified source device and records edited file on specified destination device.
EDIT File [from] <name> [to] <device> <esc>	<esc>,cEDIT F <name> <device> <esc>	Allows user editing of file from source device with the specified name and records edited file on specified destination device.
EDIT File [from] <disc file name> [to] <disc file name> <esc>	<esc>,cEDIT F <disc file name> <disc file name> <esc>	Allows user editing of file from specified disc file name and records edited file on specified new disc file name.

Table D-1. Escape Codes (Continued)

COMMAND SYNTAX <command sequence>	ESC,c <abbreviated command sequence> or ESC&p <parameters>	DESCRIPTION
EDIT File ↵	ESC,cEDIT F ↵	Allows user editing of file on previously-assigned source device and records edited file on previously-assigned destination device. (Note: same as Enable Edit.)
EDIT File <device> ↵	ESC,cEDIT F <device> ↵	Allows user editing of file on specified source device and records edited file on previously-assigned destination device.



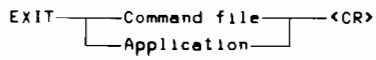
Enable/Disable Edit ↵	ESC,cE or D E ↵	Turns Edit Mode on/off.
Enable/Disable Verify ↵	ESC,cE or D V ↵ ESC&p 10C (on) ESC&p 9C (off)	Toggles Verify Mode. (Reads each record transferred and compares it with the original.)
Enable/Disable Record ↵	ESC,cE or D R ↵	In REMOTE, transfers data from computer to destination device. In LOCAL, transfers one file from DISPLAY to destination device.
Enable/Disable Numeric ↵	ESC,cE or D N ↵	Enables/disables the graphics funtions of the graphics/numeric keypad.
Enable/Disable Graphics ↵	ESC,cE or D G ↵	Enables/disables the numeric functions of the graphics/numeric keypad.
Enable Graphics ↵	ESC,cG ↵	Sets numeric/graphic keypad for graphic operation.
Enable Numeric ↵	ESC,cN ↵	Sets numeric/graphic keypad for numeric operation.



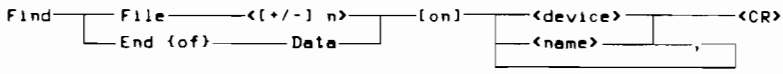
Example: Execute the list of commands in the present file on the right tape.

```
EXecute Right tape <CR>
```

EXecute <device or name> ↵	ESC,cEX <device or name> ↵	Executes commands stored on the specified device.
EXecute <disc file name> ↵	ESC,cEX <disc file name> ↵	Executes commands stored on the specified file.



EXIT Command File ↵	ESC,cEXIT C ↵	Terminates a command file.
EXITApplication ↵	ESC,cEXIT A ↵	Terminates a subsystem (e.g., BASIC)



Example: Find the seventh file on the left tape.

```
Find File 7 on Left tape <CR>
```

Example: Position the right tape forward 3 files.

```
Find File +3 on Right tape <CR>
```

Example: Find end-of-data on tape with user name "MEMO".

```
Find End of Data on MEMO <CR>
```

Find File<+, -n> [on] Left Tape ↵	ESC,cF F <+, -n> L ↵ or ESC&p (+, -n) p 1u 2C	Positions left tape to a relative (+, -n) or absolute (n) file.
Find File <+, -n> [on] Right tape ↵	ESC,cF F <+,-n> R ↵ or ESC&p (+, -n) p 2u 2C	Positions right tape to a relative (+, -n) or absolute (n) file.
Find End [of] Data [on] Left tape ↵	ESC,cF E D L ↵ or ESC&p 1u 3C	Finds end-of-data on left tape.

Table D-1. Escape Codes (Continued)

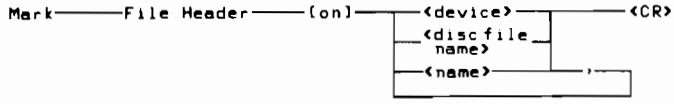
COMMAND SYNTAX command sequence	ESC,c- abbreviated command sequence or ESC&p <parameters	DESCRIPTION
Find End [of] Data [on] Right tape ↵	ESC,cF E D R ↵ or ESC&p 2u 3C	Finds end-of-data on right tape.

Example: Format a new floppy disc for use on the Terminal.
(The floppy disc is in disc unit #2.)

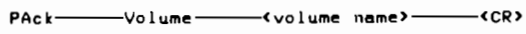
```
FORmat Volume DISC#2 <CR>
```

FORmat Volume [on] DISC# <unit> [with] Stagger <n> ↵	ESC,cFOR V DISC# <unit> S <n> ↵	Records soft sector information to make floppy disc compatible with the terminal file system.
FORmat Volume [on] DISC#< unit> ↵	ESC,cFOR V DISC# <unit> ↵	Records default soft sector information (STAGGER = 9) to make floppy disc compatible with the terminal file system.

HELLO <user.group - ↵	ESC,cHELLO <user.group> ↵	Enters user.group name to be used on shared printer listing. User and group are each limited to 8 characters.



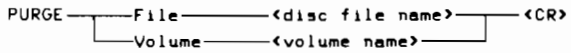
Mark File Header [on] Left tape ↵	ESC,cM F H L ↵	Writes a file mark on left tape.
Mark File Header [on] Right tape ↵	ESC,cM F H R ↵	Writes a file mark on right tape.
Mark File Header [on] EXternal printer ↵	ESC,cM F H EX ↵	Causes a Form Feed on external printer.
Mark File Header [on] SHared printer#5 ↵	ESC,cM F H SH#5 ↵	Causes a Form Feed on shared printer.
Mark File Header [on] <name> ↵	ESC,cM F H <name> ↵	Writes a file mark or causes a Form Feed on device that has a user-assigned name.
Mark File Header [on] <disc file name> ↵	ESC,cM F H <disc file name> ↵	Closes the specified file on the floppy disc.



Example: Pack a floppy disc (volume) named "DATA".

```
PAck Volume DATA <CR>
```

PAck Volume <volume name> ↵	ESC,cPA V <volume name> ↵	Deletes those files that have been purged and moves the remaining files together.



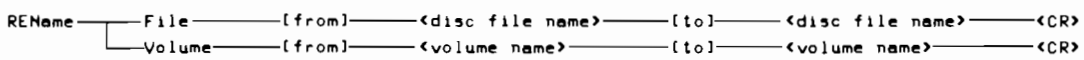
Example: Remove access to a file named "PRODUCT".

```
PURGE File PRODUCT <CR>
```

Example: Remove access to a volume named "COMPANY".

```
PURGE Volume COMPANY <CR>
```

PURGE File <disc file name> ↵	ESC,c PURGE F <disc file name> ↵	Removes file access to the specified file.
PURGE Volume <volume name> ↵	ESC,c PURGE V <volume name> ↵	Removes access to the specified volume. (Use CReate command to make floppy disc useable as a new volume.)



Example: Rename the disc file named "MONTH3" on volume "Accounting" to "MONTH4".

```
REName File from MONTH3;ACCOUNTING to MONTH4 <CR>
```

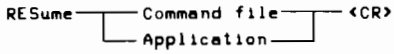
Example: Rename the floppy disc (volume) named "YEAR78" to "YEAR79".

```
REName Volume from YEAR78 to YEAR79 <CR>
```

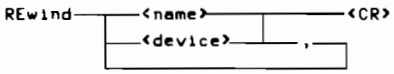
Table D-1. Escape Codes (Continued)

COMMAND SYNTAX <command sequence ·	ESC,c <abbreviated command sequence · ESC or ESC&p <parameters ·	DESCRIPTION
REName File [from] <disc file name · [to] <file name · ↵	ESC,cREN F <disc file name · <file name · ↵	Renames a disc file.
REName Volume [from] <volume name · [to] <volume name · ↵	ESC,cREN V <volume name · <volume name · ↵	Renames a floppy disc (volume).

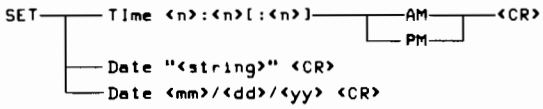
REPort Status [of] Command ↵	ESC,cREP S C ↵	Sends result of last command executed or current device state to the datacomm. if in remote. If last command was EXecute, REPort will be returned after EXecute terminates.



RESume Command file ↵	ESC,cRES C ↵	Resumes execution of the command file after a SUspend command is encountered.
RESume Application ↵	ESC,cRES A ↵	Resumes operation of a subsystem after a suspend command is encountered.



REwind Left tape ↵	ESC,cRE L ↵ or ESC&p 1u 0C	Rewinds left tape.
REwind Right tape ↵	ESC,cRE R or ESC&p 2u 0C	Rewinds right tape.
REwind <name · ↵	ESC,cRE <name · ↵	Rewinds tape that has a user-assigned name.



Example: Set time to 2:15:45 PM.

```
SET TIme 2:15:45 PM <CR>
```

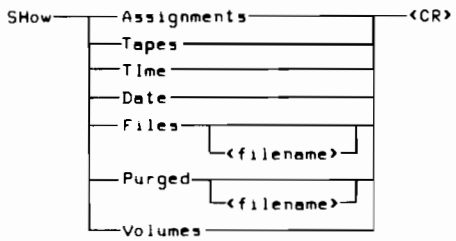
Example: Set date to Tuesday, August 8, 1978.

```
SET Date "Tuesday, August 8, 1978" <CR>
or
SET Date 8/8/78 <CR>
```

SET TIme <n ·:<n ·[:<n ·] AM (or PM) ↵	ESC,cSET TI <n ·:<n ·[:<n ·] AM or PM ↵	Sets time on terminal.
SET Date "<date string ·" ↵	ESC,cSET D "<date string ·" ↵	Sets date on terminal. (Maximum length of date string is 30 characters.
SET Date <month ·/<days ·/<year · ↵	ESC,cSET D <month ·/<day ·/<year · ↵	Sets date on terminal (terminal creates date string automatically). The date is used as the "Create Date" in file directory.
SET Program [to] program size ↵	ESC,cSET P program size ↵	Allocates common memory to application program usage. (Program size is allocated in multiples of 256).

Table D-1. Escape Codes (Continued)

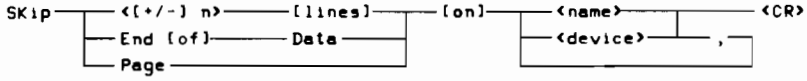
COMMAND SYNTAX <command sequence>	ESC,c <abbreviated command sequence> or ESC&p <parameters>	DESCRIPTION



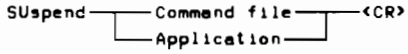
Example: List all files that have "CTUL" in their name.
SHow Files CTUL* ↵

Example: List the first nine CTU emulator files for the right tape.
SHow Files CTUR00? ↵

SHow Assignments ↵	ESC,cSH A ↵	Lists current Source, Destination, Log, CTU, Printer, device and user assigned names on Log device.
SHow Tapes ↵	ESC,cSH T ↵	Lists current tape file and remaining tape space on log device.
SHow Time ↵	ESC,cSH T ↵	Lists current time on log device.
SHow Date ↵	ESC,cSH D ↵	Lists current date on log device.
SHow Files ↵	ESC,cSH F ↵	Lists the disc file names contained on each floppy disc (volume) connected to the terminal on the log device.
SHow Purged ↵	ESC,cSH P ↵	Lists the purged disc file names contained on each floppy disc (volume) connected to the terminal on the log device.
SHow Volumes ↵	ESC,cSH V ↵	List HP-IB device addresses and floppy disc (volume) names on-line to the terminal on the log device.



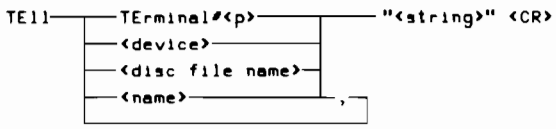
SKip <+, -n> [lines] [on] Right tape ↵	ESC,cSK <+, -n> R ↵ or ESC&p (+, -n)p 2u 1C	Positions right tape to a relative (+, -n) line.
SKip <+, -n> [lines] [on] EXternal printer ↵	ESC,cSK <+, -n> EX ↵ or ESC&p (+, -n)p 4u 1C	Positions external printer to a relative (+, -n) line (Some printers may not respond to minus sign.)
SKip <+, -n> [lines] [on] SHared printer#5 ↵	ESC,cSK <+, -n> SH#5 ↵	Positions SHared printer#5 to a relative (+, -n) line. (Some printers may not respond to minus sign.)
SKip <+, -n> [lines] [on] <name> ↵	ESC,cSK <+, -n> <name> ↵	Positions device that has user-assigned device name to a relative (+, -n) line.
SKip Page [on] EXternal printer ↵	ESC,cSK P EX ↵ or ESC&p (n)p 4u 2C	Positions external printer to a top-of-form.
SKip Page [on] SHared printer#5 ↵	ESC,cSK P SH#5 ↵	Positions shared printer to a top-of-form.
SKip End [of] Data [on] Left tape ↵	ESC,cSK E D L ↵	Positions left tape beyond end-of-data mark.
SKip End [of] Data [on] Right tape ↵	ESC,cSK E D R ↵	Positions right tape beyond end-of-data mark.



SUspend Command file ↵	ESC,cSU C ↵	Suspends execution of a command file.
SUspend Application ↵	ESC,cSU A ↵	Suspends operation of the subsystem.

Table D-1. Escape Codes (Continued)

COMMAND SYNTAX <command sequence>	ESC,c <abbreviated command sequence> or ESC&p <parameters>	DESCRIPTION



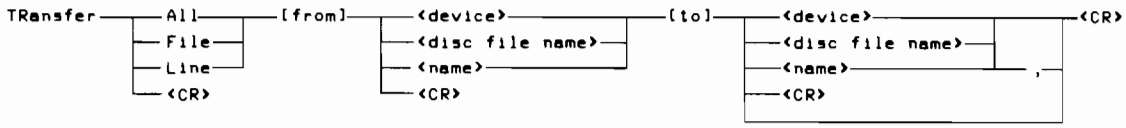
Note: Single quotes or double quotes may be used.

Example: Tell the right tape: "End of program."

```
TELL Right tape "End of Program" <CR>
```

TEll <device> "<string>"	ESC,cTE <device> "<string>"	Transfers data string to the specified device.
TEll TErminal#<p> "<string>"	ESC,cTE TE#<p> "<string>"	Transfers data string to the specified terminal in the network.
TEll <disc file name> "<string>"	ESC,cTE <disc file name> "<string>"	Transfers data string to the specified disc file.

Test	ESC,cT or ESC z	Tests terminal.
Test DAtacomm	ESC,cT DA or ESC x	Tests terminal datacomm. (Hood connector must be used.)
Test Tapes	ESC,cT T or ESC&p (1 or 2)u 7C	Tests CTUL file 1 and CTUR file 1.
Test TErminal#<p>	ESC,cT TE# <p>	Tests data path to other terminal in network.
Test Hp-ib#<m>	ESC,cT H# <m>	Tests a specific HP-IB interface PCA in terminal.
Test DISCS	ESC,cT DISCS	Tests the floppy disc units.



Example: Copy the contents of "PROGRAM1" (disc file) to "PROGRAM2" (disc file).

```
TRansfer File from PROGRAM1 to PROGRAM2 <CR>
```

Example: Copy the contents of graphics memory to disc file "TOPVIEW"

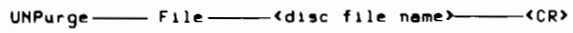
```
TRansfer File from Graphics to TOPVIEW <CR>
```

NOTE: The TRANSFER command sequence copies data in 8-bit binary form. For 7-bit ASCII transfers, use the COPY command sequence.

Transfer File [from] Source [to] Display	ESC,cTR F S D1	Transfers one file from Source device to display.
TRansfer File [from]Display [to] Destination	ESC,cTR F DI D	Transfers all data from the current cursor position in the display workspace to Destination device.
TRansfer All	ESC,cTR A	All files (current position) from previously-assigned Source device are transferred to previously-assigned Destination device.
TRansfer File	ESC,cTR F	One file (current position) from previously-assigned Source device is transferred to previously-assigned Destination device.

Table D-1. Escape Codes (Continued)

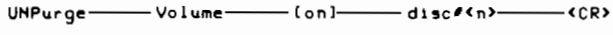
COMMAND SYNTAX <command sequence>	ESC,c <abbreviated command sequence> or ESC&p <parameters>	DESCRIPTION
TRansfer Line	ESC,cTR L	One line (current position) from previously-assigned Source device is transferred to previously-assigned Destination device.
TRansfer All [from]<device>[to]<device(s)>	ESC,cTR A <device> <device(s)>	All files (current position) from a specified device are transferred to one or more specified device(s).
TRansfer File [from]<device>[to]<device(s)>	ESC,cTR F <device> <device(s)>	One file (current position) from a specified device is transferred to one or more specified device(s).
TRansfer Line [from]<device>[to]<device(s)>	ESC,cTR L <device> <device(s)>	One line (current position) from a specified device is transferred to one or more specified device(s).
TRansfer All [from]<name>[to]<name>	ESC,cTR A <name> <name>	All files (current position) from user-assigned device are transferred to user-assigned device. One file is transferred if disc file.
TRansfer File [from]<name>[to]<name>	ESC,cTR F <name> <name>	One file (current position) from user-assigned device or disc file name is transferred to user-assigned device or disc file name.
TRansfer Line [from]<name>[to]<name>	ESC,cTR L <name> <name>	One line (current position) from user-assigned device or disc file name is transferred to user-assigned device or disc file name.



Example: Unpurge the disc file named "ORDERS".

```
UNPurge File ORDERS <CR>
```

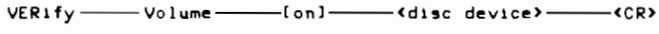
UNPurge File <disc file name> [;<volume name>]	ESC,cUNP F <disc file name> [;<volume name>]	Allows user access to a previously-purged disc file on an unpacked volume.



Example. Unpurge the diskette mounted in disc drive #1.

```
UNPURGE Volume on DISC#1 <CR>
```

UNPurge Volume [on] DISC# <n>	ESC,cUNP V DISC# <n>	Allows user access to a previously-purged diskette.



Example: Verify floppy disc (volume) on disc #2.

```
VERIFY Volume DISC#2 <CR>
```

VERify Volume [on] DISC# <n>	ESC,cVER V DISC# <n>	Reads entire volume checking CRC's and recording tolerances.

COMMUNICATIONS FLOWCHARTS

APPENDIX
E

This appendix contains reference information on terminal communication flowcharts.

The flowchart in figure E-1 illustrates the overall point-to-point communication function. The various configuration parameters (switches) are included in the diagram. Figure E-2 illustrates the way the terminal responds to various Processor PCA switches.

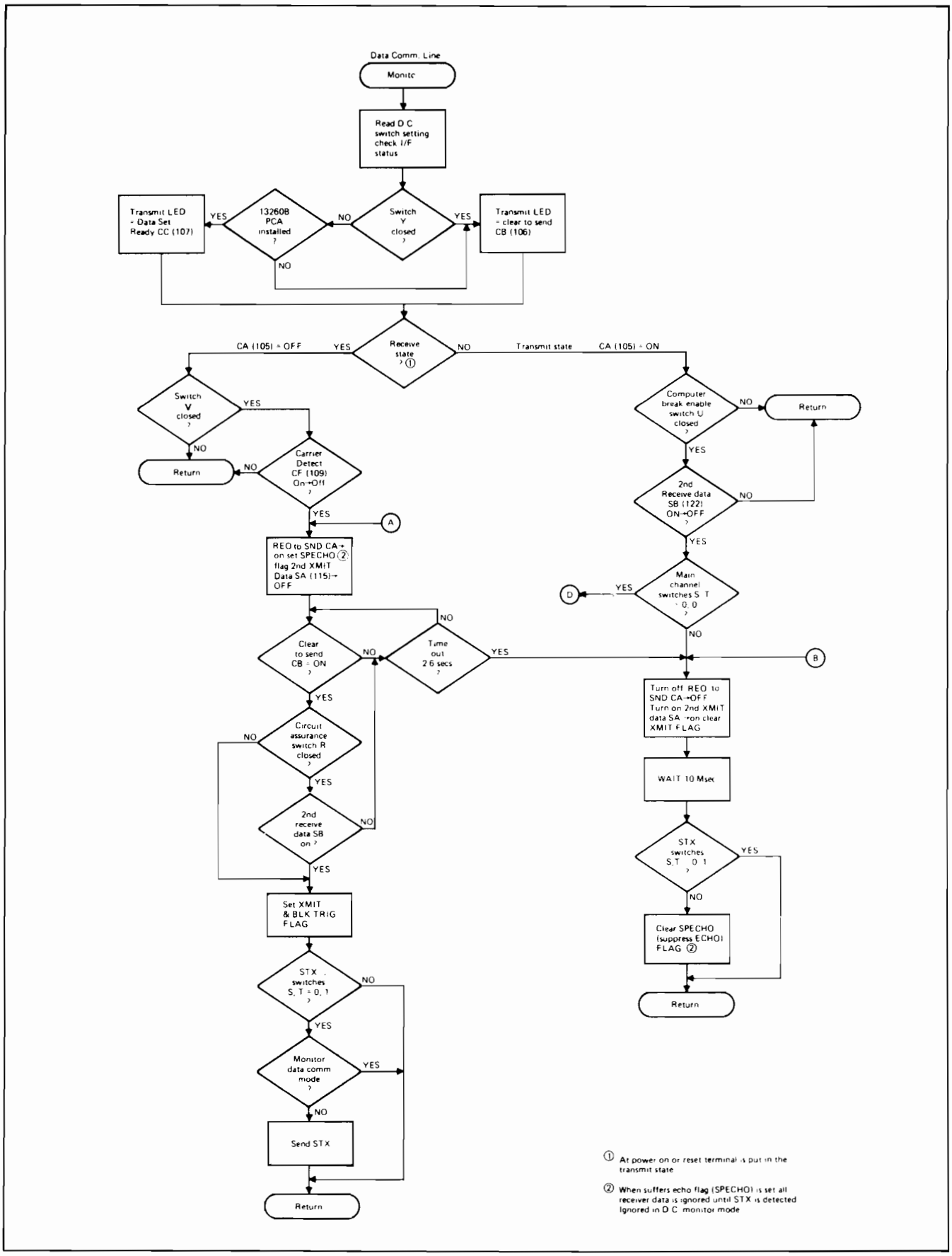


Figure E-1. Point-to-Point Communication Flowcharts (Sheet 1 of 3)

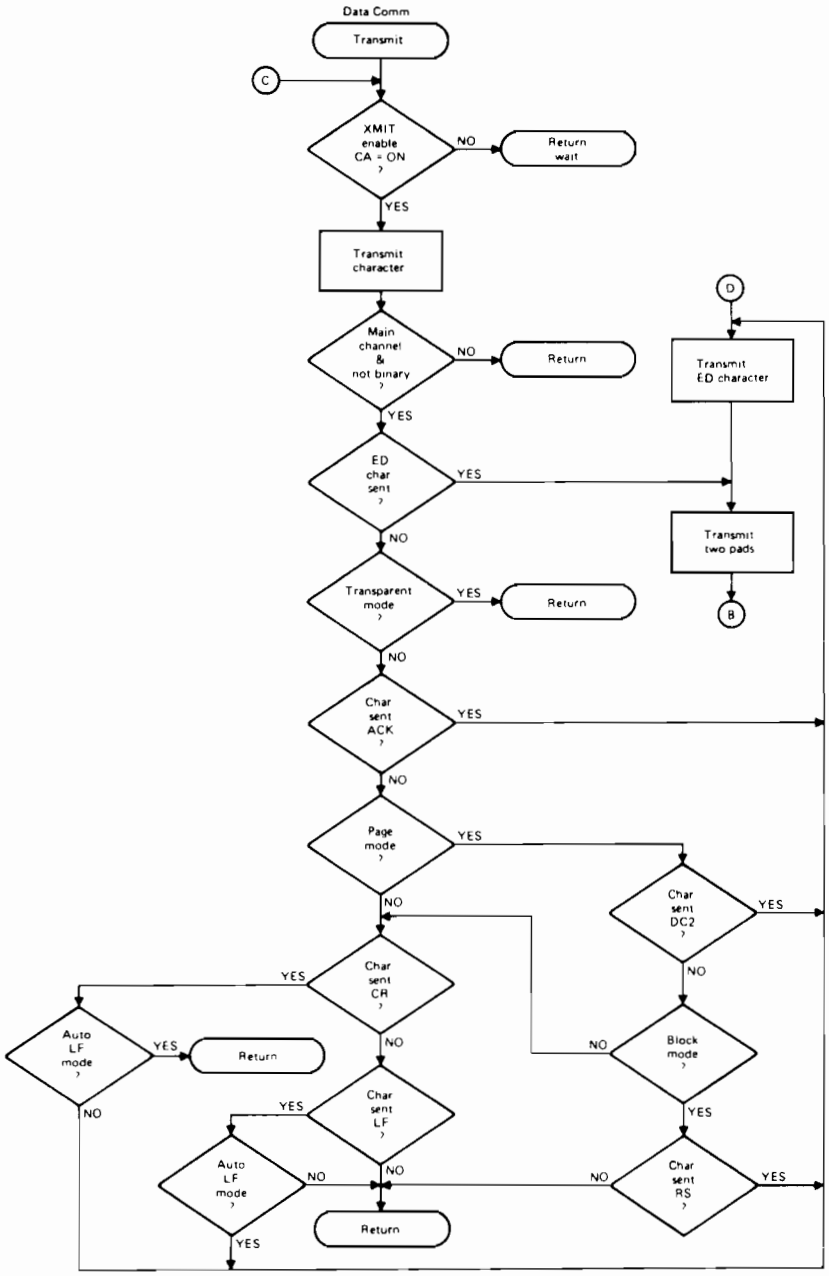


Figure E-1. Point-to-Point Communication Flowcharts (Sheet 2 of 3)

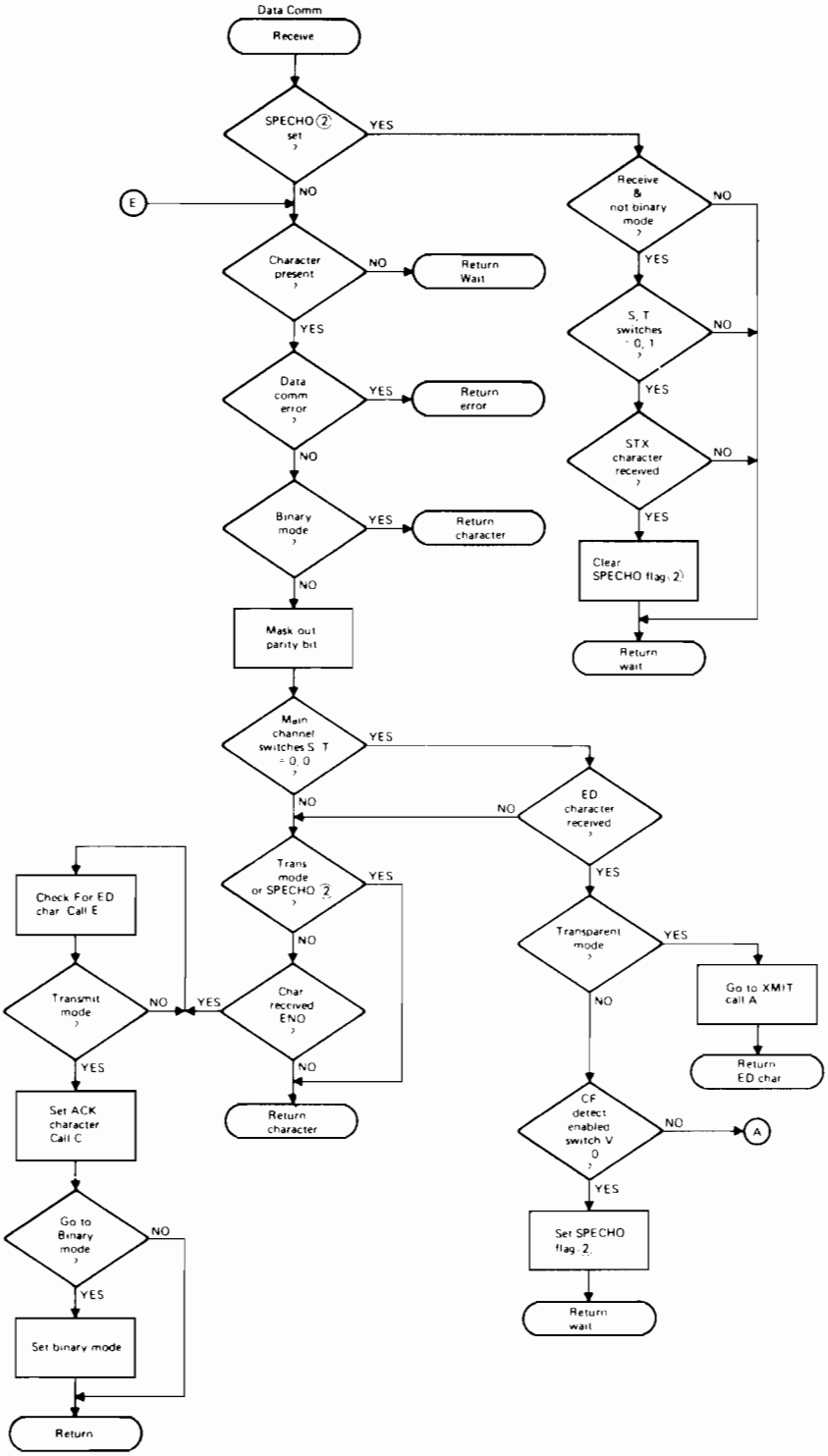


Figure E-1. Point-to-Point Communication Flowcharts (Sheet 3 of 3)

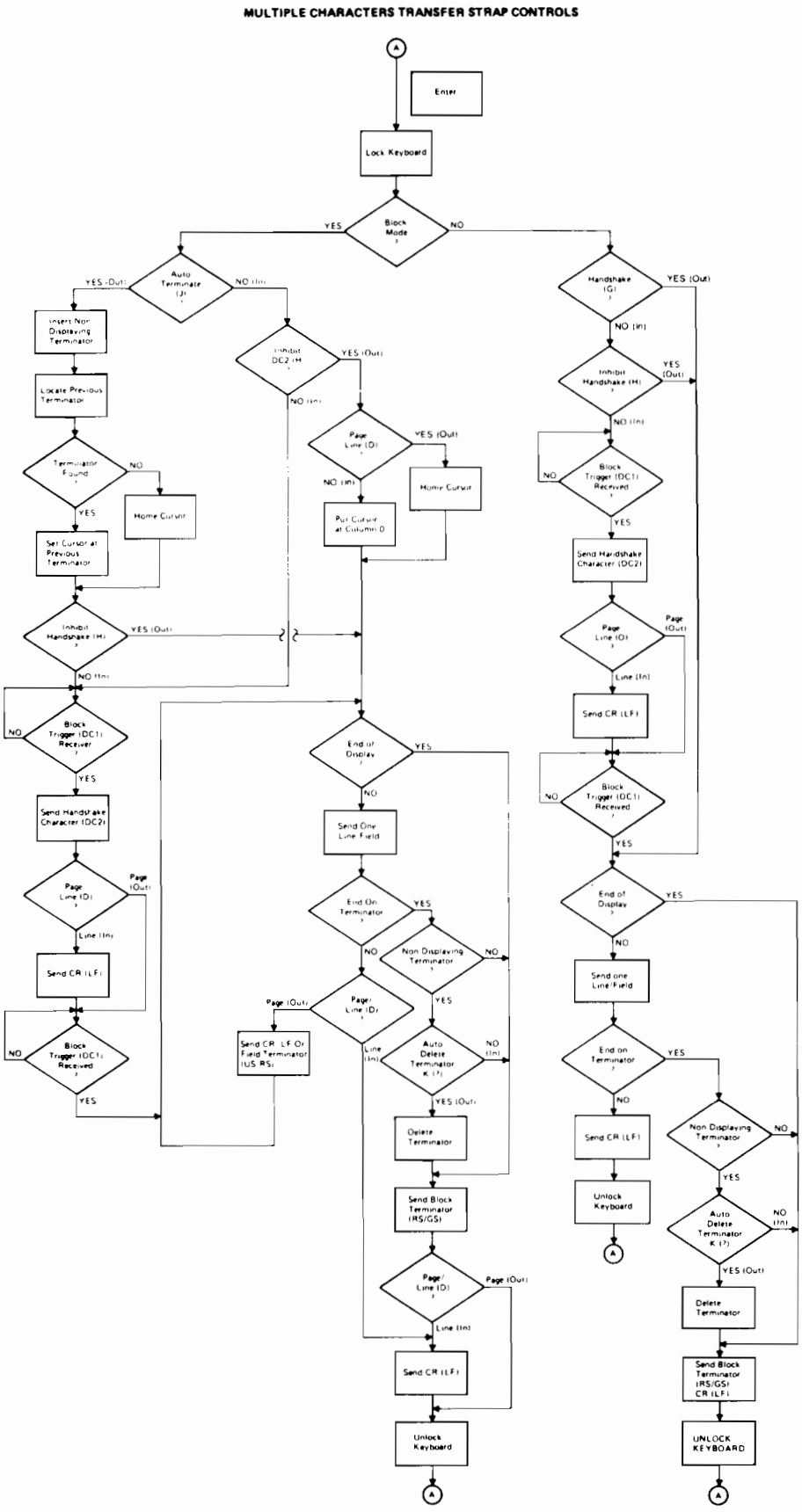


Figure E-2. Processor PCA Communication Switches Flowcharts (Sheet 1 of 4)

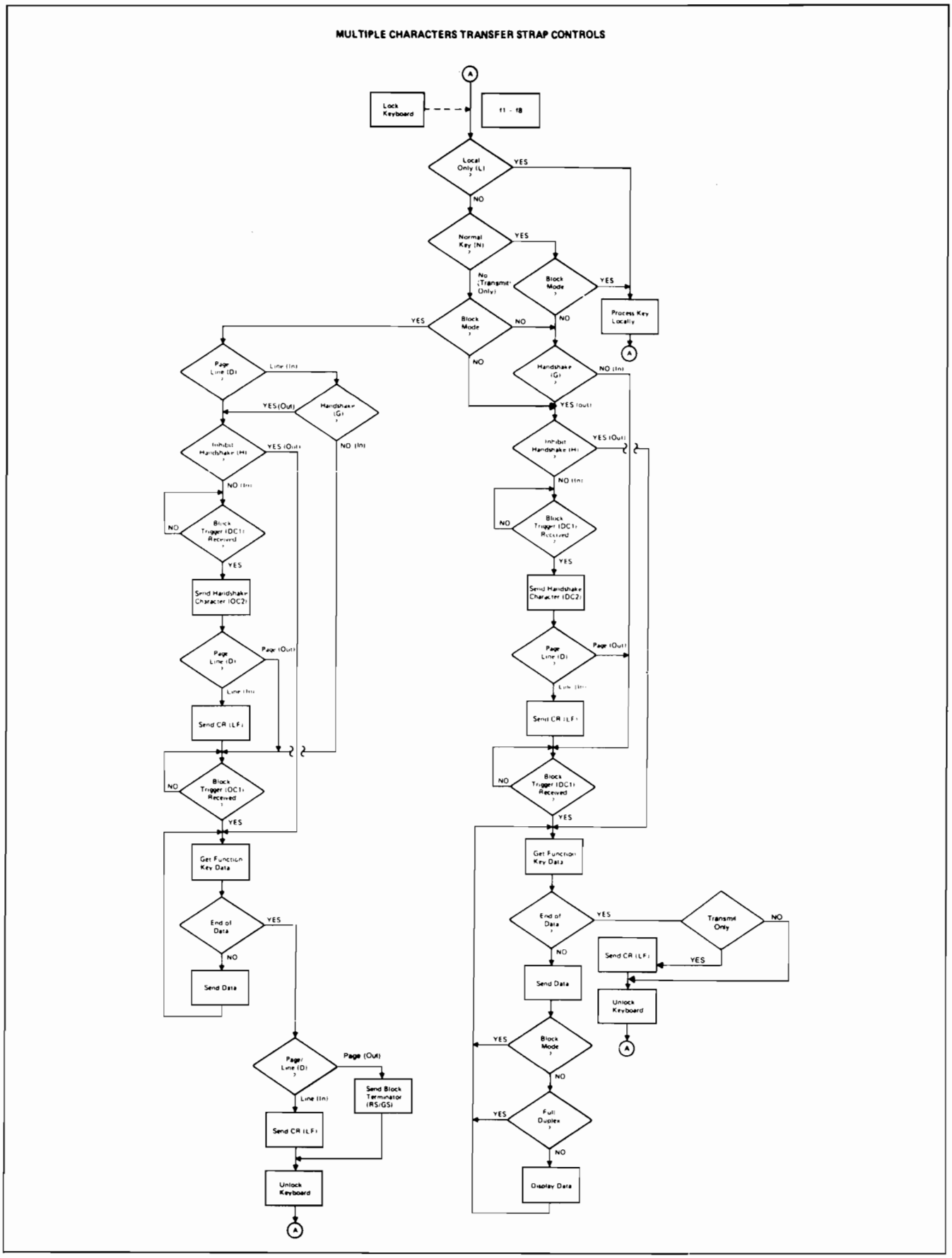


Figure E-2. Processor PCA Communication Switches Flowcharts (Sheet 2 of 4)

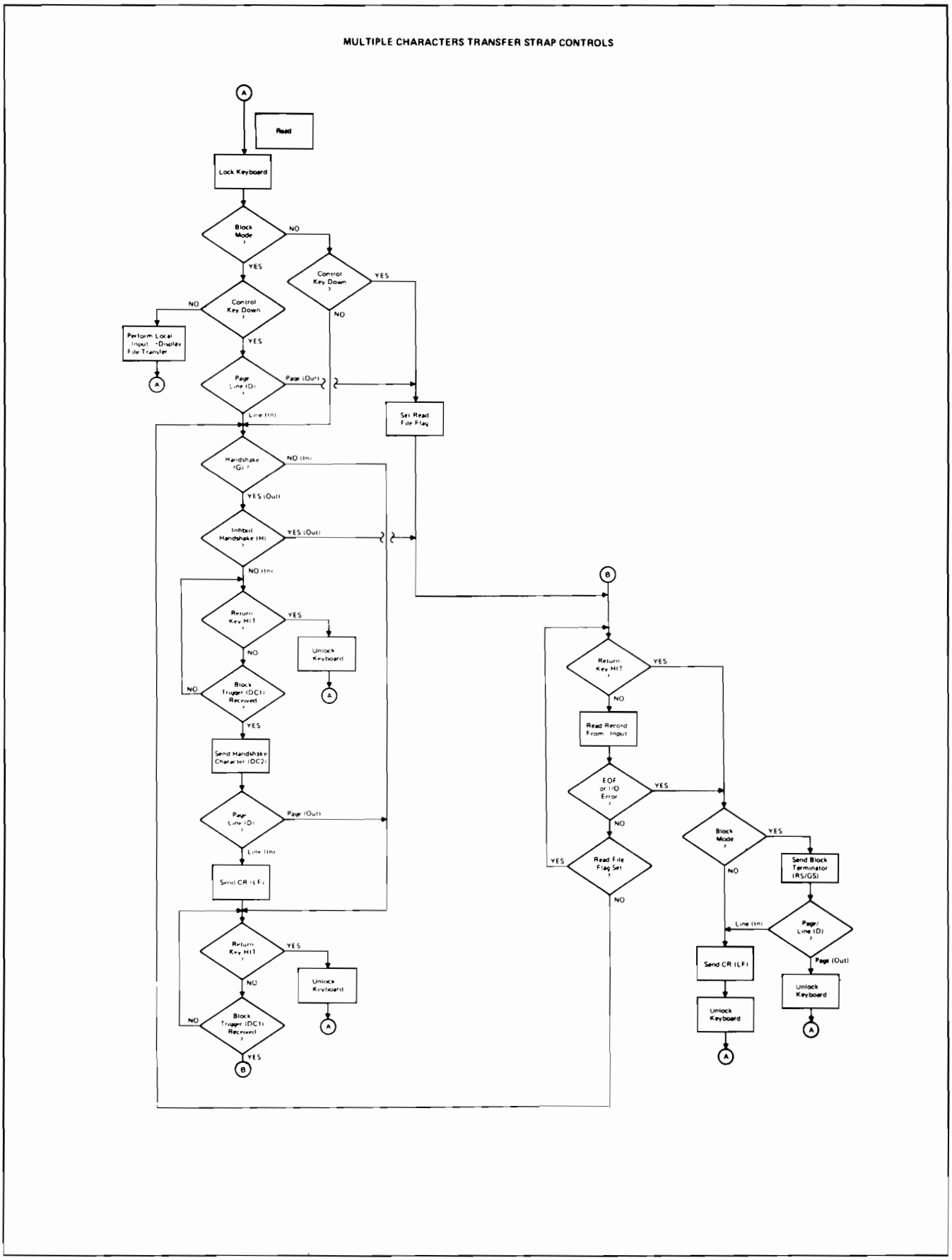


Figure E-2. Processor PCA Communication Switches Flowcharts (Sheet 3 of 4)

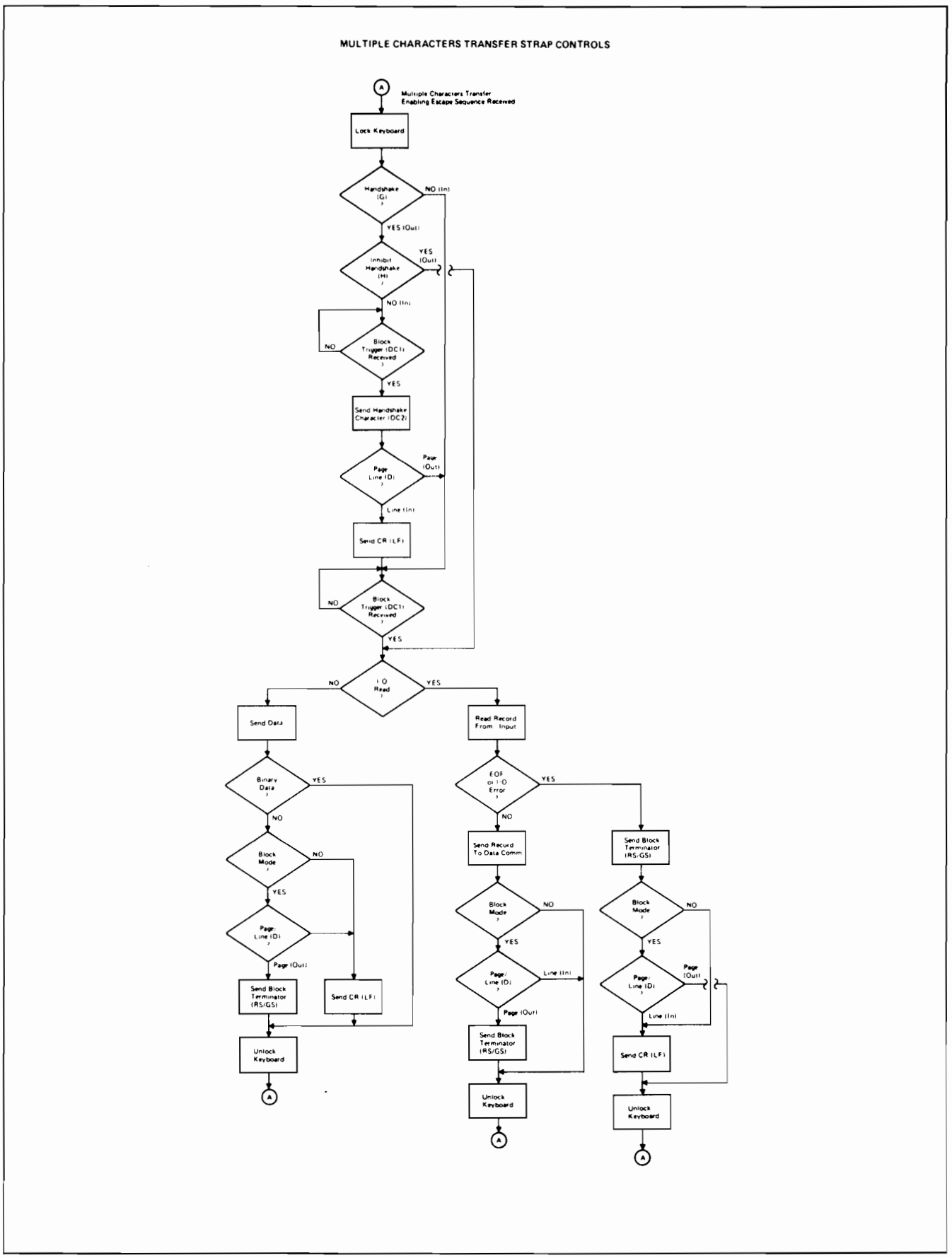


Figure E-2. Processor PCA Communication Switches Flowcharts (Sheet 4 of 4)

CHARACTER CODES

APPENDIX F

INTRODUCTION

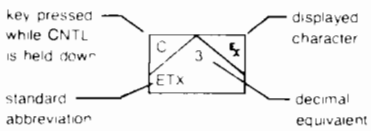
This appendix contains character code information for the ASCII, EBCDIC, and large character sets.

Table F-1. Character Code Chart Reference Tables

BIT 7 6 5 / 4 3 2 1	CONTROL (CNTL) CHARACTERS 0 0 0	CONTROL (CNTL) CHARACTERS 0 0 1	DISPLAYABLE CHARACTERS 0 1 0	DISPLAYABLE CHARACTERS 0 1 1	DISPLAYABLE CHARACTERS 1 0 0	DISPLAYABLE CHARACTERS 1 0 1	DISPLAYABLE CHARACTERS 1 1 0	DISPLAYABLE CHARACTERS 1 1 1	ESCAPE SENT FIRST 0 1 0	ESCAPE SENT FIRST 0 1 1	ESCAPE SENT FIRST 1 0 0	ESCAPE SENT FIRST 1 0 1	ESCAPE SENT FIRST 1 1 0	ESCAPE SENT FIRST 1 1 1
0000	@ 0 N_U NUL	P 16 DLE	SP	0	@	P	`	p	SP	0	@ DELAY 1 SEC	P DELETE CHAR	CURSOR RELATIVE SENSE	p t_1
0001	A 1 S_H SOH	Q 17 D_1 DC1	!	1	A	Q	a	q	!	1 A ↑ CURSOR UP	Q INSERT CHAR ON	a CURSOR ABSOLUTE SENSE	q t_2	
0010	B 2 S_X STX	R 18 D_2 DC2	"	2	B	R	b	r	"	2 CLEAR TAB	B ↓ CURSOR DOWN	R INSERT CHAR OFF	b KEYBOARD ENABLE	r t_3
0011	C 3 E_X ETX	S 19 D_3 DC3	#	3	C	S	c	s	#	3 CLEAR ALL TABS	C → CURSOR RIGHT	S ROLL UP	c KEYBOARD DISABLE	s t_4
0100	D 4 E_T EOT	T 20 D_4 DC4	$	4	D	T	d	t	$	4 SET LEFT MARGIN	D ← CURSOR LEFT	T ROLL DOWN	d ENTER	t t_5
0101	E 5 E_Q ENQ	U 21 N_K NAK	%	5	E	U	e	u	%	5 SET RIGHT MARGIN	E RESET TERMINAL	U NEXT PAGE	e BINARY READ	u t_6
0110	F 6 A_K ACK	V 22 S_Y SYN	&	6	F	V	f	v	& PARA METER SEQUENCE	6 START ALPHA FIELD	F CURSOR HOME DOWN	V PREV PAGE	f MODEM DIS CONNECT	v t_7
0111	G 7 BEL	W 23 E_B ETB	'	7	G	W	g	w	'	7 START NUMERIC FIELD	G CURSOR RETURN	W FORMAT MODE ON	g SOFT RESET	w t_8
1000	H 8 B_S BS	X 24 C_N CAN	(	8	H	X	h	x	(	8 START ALPHNUM FIELD	H HOME CURSOR (SEE NOTE 3)	X FORMAT MODE OFF	h HOME CURSOR (SEE NOTE 3)	x DATA COM SELF TEST
1001	I 9 H_T HT	Y 25 E_M EM	)	9	I	Y	i	y	DEFINE CHAR SET)	9	I HORI ZONTAL TAB	Y DISPLAY FUNCTIONS ON	i BACK TAB	y MONITOR MODE ON
1010	J 10 L_F LF	Z 26 S_B SUB	*	:	J	Z	j	z	GRAPHICS SEQUENCE *	:	J CLEAR DSPLY	Z DISPLAY FUNCTIONS OFF	j SOFT KEY DISPLAY ON	z TERMINAL SELF TEST
1011	K 11 V_T VT	[27 E_C ESC	+	;	K	[	k	{	+	;	K ERASE TO END OF LINE	[START UNPROTECT FIELD	k SOFT KEY DISPLAY OFF	{ START AMIT ONLY FIELD
1100	L 12 F_F FF	\ 28 F_S FS	,	<	L	\	l	|	,	<	L INSERT LINE	\	l MEMORY LOCK ON	|
1101	M 13 C_R CR	] 29 G_S GS	-	=	M	]	m	}	-	=	M DELETE LINE	] END UNPROTECT FIELD	m MEMORY LOCK OFF	}
1110	N 14 S_O SO	^ 30 R_S RS	.	>	N	^	n	~	.	>	N INSERT CHAR W/WRAP ON	^ TERM PRIMARY STATUS	n	~ SEND SECOND ARY STATUS
1111	O 15 S_I SI	_ 31 U_S US	/	?	O	_	o	DEL	/	?	O DELETE CHAR W/WRAP	_ INSERT NON-DISP TERMINATR	o	DEL

1. Lowercase letter, lowercase symbol, and control character codes are generated by standard terminal, but associated characters are not displayed on the screen. Press tape test key for displayable character set.
2. Single character escape sequences and control codes not listed with a function are neither acted upon nor displayed.
3. ESC H homes cursor including transmit-only fields. ESC h homes cursor excluding transmit-only fields.

Control Character Legend



A_K — ACKNOWLEDGE
BEL — BELL
B_S — BACKSPACE
C_N — CANCEL LINE
C_R — CARRIAGE RETURN
D_L — DATA LINK ESCAPE
D_1 — DEVICE CONTROL 1
D_2 — DEVICE CONTROL 2
D_3 — DEVICE CONTROL 3
D_4 — DEVICE CONTROL 4
DEL — DELETE
E_M — END OF MEDIUM
E_Q — ENQUIRY
E_T — END OF TRANSMISSION
E_C — ESCAPE
E_B — END OF BLOCK
E_X — END OF TEXT
F_F — FORM FEED
F_S — FILE SEPARATOR
G_S — GROUP SEPARATOR
H_T — HORIZONTAL TAB
L_F — LINE FEED
N_K — NEGATIVE ACKNOWLEDGE
R_S — RECORD SEPARATOR
S_I — SHIFT IN
S_O — SHIFT OUT
SP — SPACE
S_H — START OF HEADING
S_X — START OF TEXT
S_B — SUBSTITUTE
S_Y — SYNCHRONOUS IDLE
U_S — UNIT SEPARATOR
V_T — VERTICAL TAB

Table F-2. ASCII Character Set

DECIMAL VALUE	GRAPHIC	COMMENTS	ALTERNATE CHARACTER	DECIMAL VALUE	GRAPHIC	COMMENTS
0	^@	Null	@	64	@	Commercial at
1	^A	Start of heading	A	65	A	Uppercase A
2	^B	Start of text	B	66	B	Uppercase B
3	^C	End of text	C	67	C	Uppercase C
4	^D	End of transmission	D	68	D	Uppercase D
5	^E	Enquiry	E	69	E	Uppercase E
6	^F	Acknowledge	F	70	F	Uppercase F
7	^G	Bell	G	71	G	Uppercase G
8	^H	Backspace	H	72	H	Uppercase H
9	^I	Horizontal tabulation	I	73	I	Uppercase I
10	^J	Line feed	J	74	J	Uppercase J
11	^K	Vertical tabulation	K	75	K	Uppercase K
12	^L	Form feed	L	76	L	Uppercase L
13	^M	Carriage return	M	77	M	Uppercase M
14	^N	Shift out	N	78	N	Uppercase N
15	^O	Shift in	O	79	O	Uppercase O
16	^P	Data link escape	P	80	P	Uppercase P
17	^Q	Device control 1 (X-ON)	Q	81	Q	Uppercase Q
18	^R	Device control 2	R	82	R	Uppercase R
19	^S	Device control 3 (X-OFF)	S	83	S	Uppercase S
20	^T	Device control 4	T	84	T	Uppercase T
21	^U	Negative acknowledge	U	85	U	Uppercase U
22	^V	Synchronous idle	V	86	V	Uppercase V
23	^W	End of transmission block	W	87	W	Uppercase W
24	^X	Cancel	X	88	X	Uppercase X
25	^Y	End of medium	Y	89	Y	Uppercase Y
26	^Z	Substitute	Z	90	Z	Uppercase Z
27	^[	Escape	[	[1] 91	[	Opening bracket
28	^\	File separator	\	[2] 92	\	Reverse slant
29	^]	Group separator	]	[1] 93	]	Closing bracket
30	^^	Record separator	^	[1] 94	^	Circumflex
31	^_	Unit separator	_	[2] 95	_	Underscore
32		Space (Blank)		96	`	Grave accent
[1]33	!	Exclamation point		97	a	Lowercase a
34	"	Quotation mark		98	b	Lowercase b
35	#	Number sign		99	c	Lowercase c
36	$	Dollar sign		100	d	Lowercase d
37	%	Percent sign		101	e	Lowercase e
38	&	Ampersand		102	f	Lowercase f
39	'	Apostrophe		103	g	Lowercase g
40	(	Opening parenthesis		104	h	Lowercase h
41	)	Closing parenthesis		105	i	Lowercase i
42	*	Asterisk		106	j	Lowercase j
43	+	Plus		107	k	Lowercase k
44	,	Comma		108	l	Lowercase l
45	-	Hyphen (Minus)		109	m	Lowercase m
46	.	Period (Decimal)		110	n	Lowercase n
47	/	Slant		111	o	Lowercase o
48	0	Zero		112	p	Lowercase p
49	1	One		113	q	Lowercase q
50	2	Two		114	r	Lowercase r
51	3	Three		115	s	Lowercase s
52	4	Four		116	t	Lowercase t
53	5	Five		117	u	Lowercase u
54	6	Six		118	v	Lowercase v
55	7	Seven		119	w	Lowercase w
56	8	Eight		120	x	Lowercase x
57	9	Nine		121	y	Lowercase y
58	:	Colon		122	z	Lowercase z
59	;	Semicolon		[2]123	{	Opening (left) brace
60	<	Less than		[2]124	|	Vertical line
61	=	Equals		[2]125	}	Closing (right) brace
62	>	Greater than		[2]126	~	Tilde
63	?	Question mark		127		Delete

Notes: 1. The equivalent EBCDIC character uses a different graphic.
2. No equivalent character exists in EBCDIC.

Table F-3. ASCII (7-Bit) Character Codes

GRAPHIC	DEC	OCT	HEX
NUL	0	0	0
SOH	1	1	1
STX	2	2	2
ETX	3	3	3
EOT	4	4	4
ENQ	5	5	5
ACK	6	6	6
BEL	7	7	7
BS	8	10	8
HT	9	11	9
LF	10	12	A
VT	11	13	B
FF	12	14	C
CR	13	15	D
SO	14	16	E
SI	15	17	F
DLE	16	20	10
DC1	17	21	11
DC2	18	22	12
DC3	19	23	13
DC4	20	24	14
NAK	21	25	15
SYN	22	26	16
ETB	23	27	17
CAN	24	30	18
EM	25	31	19
SUB	26	32	1A
ESC	27	33	1B
FS	28	34	1C
GS	29	35	1D
RS	30	36	1E
US	31	37	1F
SP	32	40	20
!	33	41	21
"	34	42	22
#	35	43	23
$	36	44	24
%	37	45	25
&	38	46	26
'	39	47	27
(	40	50	28
)	41	51	29
*	42	52	2A
+	43	53	2B
,	44	54	2C
-	45	55	2D
.	46	56	2E
/	47	57	2F
0	48	60	30
1	49	61	31
2	50	62	32
3	51	63	33
4	52	64	34
5	53	65	35
6	54	66	36
7	55	67	37
8	56	70	38
9	57	71	39
:	58	72	3A
;	59	73	3B
<	60	74	3C
=	61	75	3D
>	62	76	3E
?	63	77	3F

GRAPHIC	DEC	OCT	HEX
@	64	100	40
A	65	101	41
B	66	102	42
C	67	103	43
D	68	104	44
E	69	105	45
F	70	106	46
G	71	107	47
H	72	110	48
I	73	111	49
J	74	112	4A
K	75	113	4B
L	76	114	4C
M	77	115	4D
N	78	116	4E
O	79	117	4F
P	80	120	50
Q	81	121	51
R	82	122	52
S	83	123	53
T	84	124	54
U	85	125	55
V	86	126	56
W	87	127	57
X	88	130	58
Y	89	131	59
Z	90	132	5A
[	91	133	5B
\	92	134	5C
]	93	135	5D
^	94	136	5E
_	95	137	5F
`	96	140	60
a	97	141	61
b	98	142	62
c	99	143	63
d	100	144	64
e	101	145	65
f	102	146	66
g	103	147	67
h	104	150	68
i	105	151	69
j	106	152	6A
k	107	153	6B
l	108	154	6C
m	109	155	6D
n	110	156	6E
o	111	157	6F
p	112	160	70
q	113	161	71
r	114	162	72
s	115	163	73
t	116	164	74
u	117	165	75
v	118	166	76
w	119	167	77
x	120	170	78
y	121	171	79
z	122	172	7A
{	123	173	7B
|	124	174	7C
}	125	175	7D
~	126	176	7E
■	127	177	7F

Table F-4. EBCDIC Character Codes

GRAPHIC	DEC	OCT	HEX
NUL	0	0	0
SOH	1	1	1
STX	2	2	2
ETX	3	3	3
PF	4	4	4
HT	5	5	5
	6	6	6
DEL	7	7	7
	8	10	8
	9	11	9
	10	12	A
VT	11	13	B
FF	12	14	C
CR	13	15	D
SO	14	16	E
SI	15	17	F
DLE	16	20	10
DC1	17	21	11
DC2	18	22	12
TM	19	23	13
RES	20	24	14
NL	21	25	15
BS	22	26	16
IL	23	27	17
CAN	24	30	18
EM	25	31	19
CC	26	32	1A
CU1	27	33	1B
IFS	28	34	1C
IGS	29	35	1D
IRS	30	36	1E
IUS	31	37	1F
DS	32	40	20
SOS	33	41	21
FS	34	42	22
	35	43	23
BYP	36	44	24
LF	37	45	25
ETB	38	46	26
ESC	39	47	27
	40	50	28
	41	51	29
SM	42	52	2A
CU2	43	53	2B
	44	54	2C
ENQ	45	55	2D
ACK	46	56	2E
BEL	47	57	2F
	48	60	30
	49	61	31
SYN	50	62	32
	51	63	33
PN	52	64	34
RS	53	65	35
UC	54	66	36
EOT	55	67	37
	56	70	38
	57	71	39
	58	72	3A
CU3	59	73	3B
DC4	60	74	3C
NAK	61	75	3D
	62	76	3E
SUB	63	77	3F

GRAPHIC	DEC	OCT	HEX
SP	64	100	40
	65	101	41
	66	102	42
	67	103	43
	68	104	44
	69	105	45
	70	106	46
	71	107	47
	72	110	48
	73	111	49
	74	112	4A
.	75	113	4B
<	76	114	4C
(	77	115	4D
+	78	116	4E
ßß	79	117	4F
&	80	120	50
	81	121	51
	82	122	52
	83	123	53
	84	124	54
	85	125	55
	86	126	56
	87	127	57
	88	130	58
	89	131	59
!	90	132	5A
$	91	133	5B
*	92	134	5C
)	93	135	5D
;	94	136	5E
¬	95	137	5F
-	96	140	60
/	97	141	61
	98	142	62
	99	143	63
	100	144	64
	101	145	65
	102	146	66
	103	147	67
	104	150	68
	105	151	69
¦	106	152	6A
,	107	153	6B
%	108	154	6C
_	109	155	6D
>	110	156	6E
?	111	157	6F
	112	160	70
	113	161	71
	114	162	72
	115	163	73
	116	164	74
	117	165	75
	118	166	76
	119	167	77
	120	170	78
	121	171	79
:	122	172	7A
#	123	173	7B
@	124	174	7C
'	125	175	7D
=	126	176	7E
"	127	177	7F

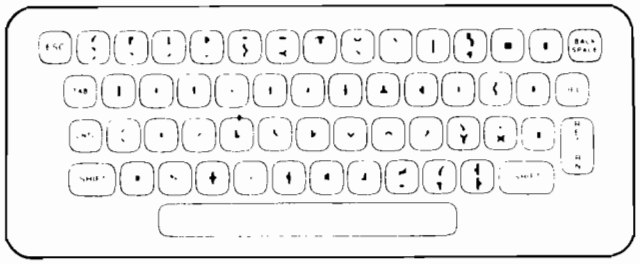
Table F-4. EBCDIC Character Codes (Continued)

GRAPHIC	DEC	OCT	HEX
	128	200	80
a	129	201	81
b	130	202	82
c	131	203	83
d	132	204	84
e	133	205	85
f	134	206	86
g	135	207	87
h	136	210	88
i	137	211	89
	138	212	8A
	139	213	8B
	140	214	8C
	141	215	8D
	142	216	8E
	143	217	8F
	144	220	90
j	145	221	91
k	146	222	92
l	147	223	93
m	148	224	94
n	149	225	95
o	150	226	96
p	151	227	97
q	152	230	98
r	153	231	99
	154	232	9A
	155	233	9B
	156	234	9C
	157	235	9D
	158	236	9E
	159	237	9F
	160	240	A0
~	161	241	A1
s	162	242	A2
t	163	243	A3
u	164	244	A4
v	165	245	A5
w	166	246	A6
x	167	247	A7
y	168	250	A8
z	169	251	A9
	170	252	AA
	171	253	AB
	172	254	AC
[	173	255	AD
	174	256	AE
	175	257	AF
	176	260	B0
	177	261	B1
	178	262	B2
	179	263	B3
	180	264	B4
	181	265	B5
	182	266	B6
	183	267	B7
	184	270	B8
	185	271	B9
	186	272	BA
	187	273	BB
	188	274	BC
]	189	275	BD
	190	276	BE
	191	277	BF

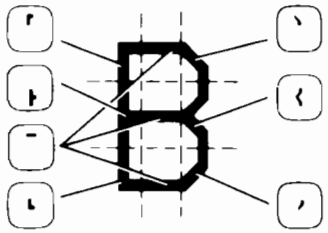
GRAPHIC	DEC	OCT	HEX
{	192	300	C0
A	193	301	C1
B	194	302	C2
C	195	303	C3
D	196	304	C4
E	197	305	C5
F	198	306	C6
G	199	307	C7
H	200	310	C8
I	201	311	C9
	202	312	CA
	203	313	CB
	204	314	CC
	205	315	CD
	206	316	CE
	207	317	CF
}	208	320	D0
J	209	321	D1
K	210	322	D2
L	211	323	D3
M	212	324	D4
N	213	325	D5
O	214	326	D6
P	215	327	D7
Q	216	330	D8
R	217	331	D9
	218	332	DA
	219	333	DB
	220	334	DC
	221	335	DD
	222	336	DE
	223	337	DF
\	224	340	E0
	225	341	E1
S	226	342	E2
T	227	343	E3
U	228	344	E4
V	229	345	E5
W	230	346	E6
X	231	347	E7
Y	232	350	E8
Z	233	351	E9
	234	352	EA
	235	353	EB
	236	354	EC
	237	355	ED
	238	356	EE
	239	357	EF
0	240	360	F0
1	241	361	F1
2	242	362	F2
3	243	363	F3
4	244	364	F4
5	245	365	F5
6	246	366	F6
7	247	367	F7
8	248	370	F8
9	249	371	F9
	250	372	FA
	251	373	FB
	252	374	FC
	253	375	FD
	254	376	FE
	255	377	FF

Table F-5. Coding the Large Character Set

The elements of the Large Character Set are associated with the keyboard as pictured below:



Each large character is actually made up of nine character segments. An example of constructing the letter "B" using the Large Character Set follows:



!	0 S	—	%&,	9	!&+ G&? G&L	E	"&, /& F&,	Q	!&+ 0 0 G&N	]	%. 0 %M	i	E	u	FM
"		.	Z	:	Z Z	F	"&, /& E	R	"&+ &@ E E	^	K	j	GL	v	GL
#	CC CC	÷	%&, 4	;	Y L	G	!&+ 0 . G&L	S	!&+ G&+ G&L	_	^^^	k	EE	w	HO
$	!C+ GC+ GCL	0	!&+ 0 0 G&L	<	3 2	H	/&? E E	T	%', 0 E	`	'	l	0 E	x	5@ EE
%	P P 3<D Y Y	1	0 E	=	%&, %&,	I	0 I	U	0 0 G&L	a	!. GM	m	$- EE	y	G?
&	!+ 5IC G&L	2	!&+ !&L F&,	>	D	J	0 L	V	0 0 GJD	b	/+ FL	n	"+ EE	z	%. F,
'		3	!&+ &@ G&L	?	!&+ >D S	K	/6A E E	W	090 HKD	c	!, G,	o	!+ GL	{	!, @ G,
(	!, 0 G,	4	F&C E	@	!&+ !.0 GIL	L	0 F&,	X	1:A E E	d	!? GM	p	"+ /L	|	Q U
)	%+ 0 %L	5	"&, F&+ G&L	A	!&+ /&? E E	M	$(- 070 E E	Y	2;D E	e	!+ G,	q	!. G?	}	%+ 5 %L
*	1:A	6	!&+ /&+ G&L	B	"&+ /&@ F&L	N	$)@ 08B E E	Z	"&. 3<D F&M	f	!+ C E	r	!, E	~	!&L
+	%C, 4	7	%&. >D E	C	!&+ 0 G&L	O	"&. 0 0 F&M	[	", 0 F,	g	!. G?	s	!, %L		
,	L	8	!&+ 5&@ G&L	D	"&+ 0 0 F&L	P	"&+ /&L E	\	@ 2:) E	h	/+ EE	t	C GL		

INDEX

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

R

S

T

U

V

W

2647F REFERENCE MANUAL

Part No. 02647-90037 Printed: 4/82

Your Comments, Please .....

Your comments assist us in improving the usefulness of our publications. Comments are an important part of the inputs used in preparing updates to manuals.

Please complete the questionnaire, fold it up and return it to us. Feel free to mark more than one box to a question and to make any additional comments. If you prefer not to give us your name just leave the last part, name and address, blank. All comments and suggestions become the property of HP.

Thank you for your help.

1. Did you have any difficulty in understanding or applying the material presented in this manual?

☐ None ☐ Minimal Difficulty ☐ Difficulty ☐ Considerable Difficulty

If so:

fold — *fold*

a. What were the "difficult" areas?

☐ Indexing? ☐ Depth of coverage? ☐ Examples?
☐ Organization? ☐ Omitted information? ☐ Other (Please explain)

b. How could we clarify these areas? ______________________________

____

2. What errors or misleading information have you found in the manual? (*Please give page numbers.*)

____

3. What was your level of programming knowledge before you started using this manual?

☐ None ☐ Beginner ☐ Intermediate ☐ Expert

4. What is your major application of the terminal?

☐ Business Adminstration ☐ Interfacing ☐ Medicine
☐ General Computation ☐ Education What kind?
☐ Engineering ☐ Data Entry ☐ Other
What kind? ☐ Control

fold — *fold*

5. What best describes how you use the terminal?

☐ Run programs written by others ☐ Do applications programming ☐ Hobbyist
☐ Write simple programs ☐ Write complex programs ☐ Other
☐ Do simple calculations

6. General Comments: ______________________________

____

____

____

Name: ______________________________

Address: ______________________________

Thank you for your help. No postage necessary if mailed in the U.S.A.

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 506 CUPERTINO, CA., U.S.A.

POSTAGE WILL BE PAID BY ADDRESSEE

Hewlett-Packard
Attn: Marketing Manager
974 Arques Avenue
Sunnyvale, California USA 94086