2000 SERIES

INDUSTRIAL WORKSTATIONS

98464-001B

Printed in the United States of America
Part Number 98464-001B

XYCOM
750 North Maple Road
Saline, Michigan 48176
(313) 429-4971

XYCOM REVISION RECORD

Revision	*Description*	*Date*
A	Manual Released	3/92
B	Incorporated PCN's 159 and 162	7/93

Address comments concerning this manual to:

xycom®
Technical Publications Dept.
750 North Maple Road
Saline, Michigan 48176
(313) 429-4971

Part Number: ***98464-001B***

XYCOM MANUAL BUG REPORT

We have provided this form to track errors that may exist in our manuals and to incorporate improvements. Please describe in the space below any errors found in this manual or any helpful suggestions to improve its usefulness. To return this form, fold it so the postage-paid* side shows, and tape it closed. We appreciate your input and will incorporate the changes in future revisions.

Current Information

Page number(s): _____ Figure number(s): _____

Information as currently printed: _____

_____

_____

_____

_____

Proposed Change(s)

Information as it should be printed: _____

_____

_____

_____

_____

Any additional information: _____

_____

_____

_____

_____

Address (optional)

Name _____

Title _____

Company _____

Address _____

City _____ State _____ ZIP _____

Telephone _____

Thank you for your time and effort.

Xycom Use Only			Log Number
Date Received:			
Date Resolved:			
Manner Resolved:	PCN #:	Revision:	Void:

**If mailing this card from outside the United States, please use an envelope with appropriate postage.*

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

First Class Permit No. 42 Saline, Michigan

POSTAGE WILL BE PAID BY ADDRESSEE

XYCOM, INC.
750 N. MAPLE ROAD
SALINE, MICHIGAN 48176

®

Xycom Manual Bug Report

Thank you for returning this form. We appreciate your feedback.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

WARNING

Dangerous voltages are present within all Xycom Industrial Terminals. These voltages will linger after all electrical power is turned off. Use caution whenever the unit is opened. Avoid touching high-voltage areas within the terminal. Do not work alone.

WARNING

The FRAGILE Cathode Ray Tube (CRT) is exposed when the front panel is opened. Wear safety glasses to protect eyes in case of accidental breakage of the CRT. Internal coating of CRT is extremely TOXIC. If exposed, RINSE IMMEDIATELY and consult a physician.

1.1 INTRODUCTION

The Xycom 2000 Series Industrial Workstations are display terminals specifically designed to perform reliably under the extreme conditions of shock, vibration, temperature, and humidity found on a plant floor. It combines many standard desktop terminal features, comprehensive character oriented graphics, and a flexible configuration.

The Xycom 2000 Series Industrial Workstations feature a CRT protected by an impact-resistant Lexan shield. Designed to seal to both NEMA 4 and NEMA 12 standards, the workstation can be mounted in a standard 19" rack with optional rack mount adaptor or in an equipment enclosure panel. This ruggedness and flexibility make the Xycom 2000 Industrial Workstation an ideal industrial interface in such applications as machine and process control, on-site data entry, and system diagnosis. A touch screen option is also available for the 2000 Series Industrial Workstations. This option allows the user to transmit codes or screens from the touch screen. The touch screen is a clear panel superimposed on the Lexan shield that divides the screen into 80 zones.

The 2000 Series features two RS-232C or RS-485 configurable serial ports, a parallel port (configurable for input or Centronics compatible output), a matrix parallel keyboard port, a standard keyboard port. It can accommodate one of several keyboards: a full-stroke PC/AT or XT style keyboard, a sealed alphanumeric keyboard with 20 function keys, or a sealed 58-key keypad with function keys.

The 2000 Industrial Workstations are equipped with 60 Kbytes of RAM memory. The OIL firmware provides 124 Kbytes of RAM memory (expandable to 251 Kbytes with the 2112 XT adapter module). All Workstations are also equipped with the 2000-01 (base terminal) firmware, which allows Hazeltine 1500 or ANSI-compatible intelligent terminal emulation (ANSI Standard X3.64). When configured as ANSI-compatible, the 2000 provides DEC VT100/220 emulation. With the 2000-01 firmware the 2000 can execute special remote commands transmitted by the host computer or stored in the 2000's own memory. These commands can format the screen and draw a variety of figures, such as lines, boxes, and high-resolution bar graphs.

The unique, open-ended design of the 2000 allows the user to configure the workstation in many different ways. This manual supports the base (terminal emulation) version of the 2000 Industrial Workstation. Xycom produces a variety of firmware upgrades to adapt the workstation to virtually any environment or function. Firmware upgrades that contain OIL (Xycom's easy-to-use Operator Interface Language), Softscreen™, and PLC (Programmable Logic Controller) communications are not addressed in this manual. For more information on the 2000 Series Industrial Workstations with OIL, and Softscreen™, contact your Xycom representative.

1.2 MANUAL STRUCTURE

A brief overview of the content of the chapters in this manual is shown below.

Chapter 1	**Introduction**: an overview of the 2000 Series Industrial Workstations, including functional and environmental specifications
Chapter 2	**Installation**: describes components of the 2000 Series Industrial Workstations and give instructions for configuring hardware
Chapter 3	**Basic Concepts**: an introduction to basic programming and application concepts of the 2000 Series Industrial Workstations, including a description of the menu structure
Chapter 4	**Keyboards and Touch Screen**: codes for the various keyboards that may be used with the 2000 Industrial Workstation and an overview of the optional touch screen
Chapter 5	**Video Display**: information on the video display, including graphics, characters, and manipulating images
Chapter 6	**Remote Commands**: instructions for using remote commands
Chapter 7	**Communications**: information on the communications capability of the 2000 Series Industrial Workstations
Appendix A	**Mounting Dimensions**: instructions for rack- or panel-mounting the 2000 Series Industrial Workstations
Appendix B	**Process Graphics Chart**: lists all 256 graphics characters, in all modes
Appendix C	**VT100/200 Codes Not Supported**: lists unsupported codes
Appendix D	**Quick Reference Guide**: lists connector pinouts, jumper settings, and switch settings
Appendix E	**Front Panel Keypad**: instructions for using the front panel keypad on the 2005 and 2050/2060 versions of the workstations

1.3 SPECIFICATIONS

Table 1-1. 2000 Series Industrial Workstations Specifications

CHARACTERISTIC	SPECIFICATION			
Mechanical	**2000**	**2005**	**2060**	**2050**
Dimensions Height Width Depth	 11" 13" 11"	 12.2" 16.25" 10.13"	 12.2" 19" 15.25"	
Weight	15 lbs.	22 lbs.	44 lbs.	
Mounting	panel or optional 19" rack mount using adapter			
CRT	9" amber		12" amber	12" color
	10x12 cells			8x10 cells
	25 rows x 80 standard characters			
Serial Ports (2)	RS-232C or RS-485, optically isolated			
Parallel Port	selectable input or output (output mode is Centronics compatible)			
Memory	60 Kbytes battery-backed screen RAM, (124 Kbytes with OIL firmware and expandable 251 Kbytes with the 2112 XT adapter module; 96 Kbytes with Softscreen; 224 Kbytes with 2000-MEM) socket available for 128 K x 8 EPROM chip, (for use with OIL firmware only), 96 Kbyte usuable EPROM for OIL program storage			
Electrical				
Power Supply	90-250 VAC @ 47-63 HZ (auto ranging)			115/230 VAC @ 47-63 HZ

Table 1-1. 2000 Series Industrial Workstations Specifications (cont.)

CHARACTERISTIC	SPECIFICATION			
Electrical	**2000**	**2005**	**2060**	**2050**
Fuse	1.5 Amp. 250 V Slo-Blo			
Battery	3.6 V, 1.8 Amp. hours			
Battery Life	10 years @ 25° C, typ. 3 years @ 65° C, min.			
Environmental				
Temperature Operating Non-operating	0° to 50° C -40° to 65° C			0 to 50 -20 to +65
Relative Humidity	5 - 90%, non-condensing			10 - 90%
Shock Operating Non-operating	15 g peak acceleration (11 msec duration) 30 g peak accelcraton (11 msec duration)			
Vibration Operating Non-operating	.006" peak-to-peak, 1 g max .015" peak-to-peak, 2.5 g max			

2.1 INTRODUCTION

This chapter describes the various components of the Xycom 2000 Series Industrial Workstations and includes instructions for configuring the hardware before power-up. The workstation consists of a CRT assembly, a power supply board, a single CRT controller board, and a firmware chip.

2.2 SYSTEM REQUIREMENTS

The minimum recommended hardware components needed to operate the 2000 are:

- The base terminal
- One firmware chip, OIL or SoftScreen direct connects
- A full-stroke PC/AT or XT keyboard

  or
- A sealed membrane keyboard

Although several different keyboards and keypads may be used with the 2000, a full alphanumeric key set is required for programming.

The following sections discuss and illustrate the installation of components for the 2000. The user should become familiar with all the components before beginning installation or operation.

2.3 FRONT PANEL

The 2000 is equipped with a NEMA 4/NEMA 12 sealed front panel. This panel protects the interior of the system from harsh environmental conditions whenever the system is properly panel-mounted. No ports or keypads are located on the front panel. (A front-panel keypad is available on the 2005 and 2050/2060. See Appendix E for more information.)

2.4 BACK PANEL

The back panel of the 2000 offers access to the ports, the power assembly, jumpers, switches, and the firmware chip. On the following page Figure 2-1 shows the location of these features on the back panel:

Power Switch	controls the power flow to the terminal
Power Receptacle	connects the power cord
Fuse Receptacle	contains the terminal fuse behind a plastic end cap
Jumpers	configure the ports as either RS-232 or RS-485
Switches	configure the system for specific keyboards, keypads, and touch screen options
Serial Ports (2)	for PLC and other peripheral connections; 9-pin D-type interface
Printer Port	for printer connection or data input; 25-pin female D-type interface
Parallel KYBD	for matrix-type sealed keyboard connection; 26 pin male interface
Serial KYBD	for PC/AT or XT style keyboard connection
Firmware Socket	for firmware installation

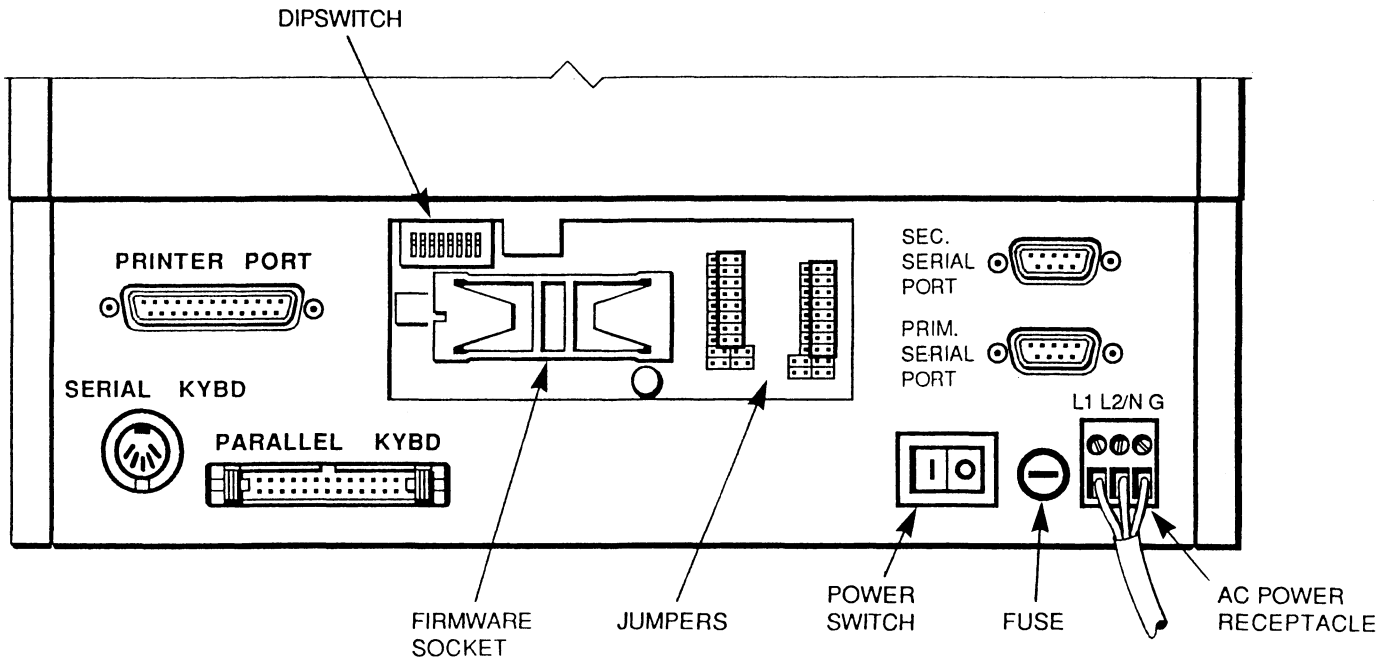


Figure 2-1. 2000 Back Panel

2.5 CONTROLLER BOARD

Normally the 2000 Industrial Workstation user should not need to remove the back panel of the unit to access the controller board. Access to the jumpers and sockets of the controller board is necessary if installing 128K x 8 EPROM into U25. See Figure 2-2 below for optional EPROM installation.

> **WARNING**
> Never attempt to open any piece of equipment without disconnecting all external power sources.

Figure 2-2, below, shows the location of the jumpers, connectors, sockets and the optional EPROM on the controller board.

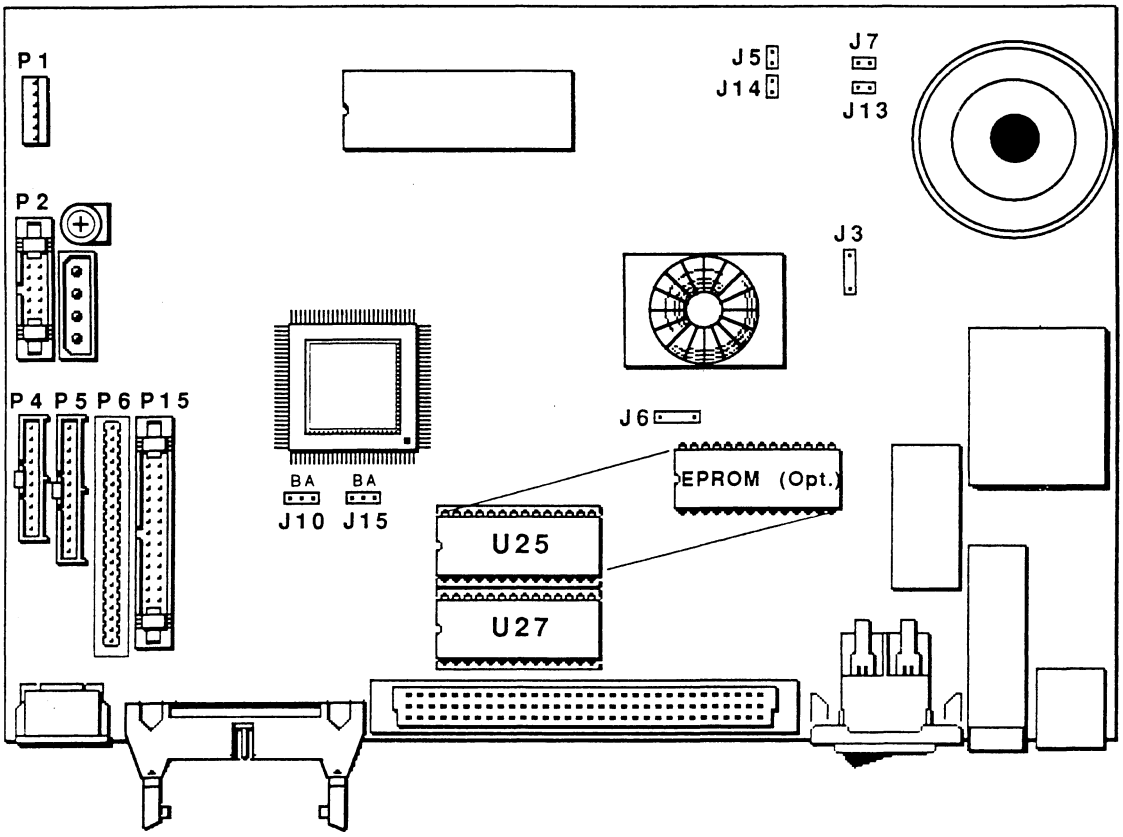


Figure 2-2. 2000 Controller Board

2.5.1 Controller Board Jumpers

Eight jumpers are located on the 2000 controller board. For locations of these jumpers, refer to Figure 2-2. Default settings are listed in Table 2-1 below.

Table 2-1. Controller Board Jumper Default Positions

Jumper	Default
J3	OUT*
J5	IN
J6	OUT*
J7	IN
J10	A
J13	IN
J14	IN
J15	B

* permanently wired positions

J10A = OIL Programs in RAM
J10B = OIL Programs in EPROM (128K x 8 EPROM installed into U25)
J13 = Beeper Enable/Disable - Remove to Disable Beeper

2.6 INSTALLATION

2.6.1 Firmware Installation

All 2000 Industrial Workstations are shipped with the 2000-01 (base terminal) firmware installed. If using the 2000 Industrial Workstation as a base terminal, proceed to the next section. If installing an optional firmware chip, follow the instructions below.

> **WARNING**
> Never attempt to open any piece of equipment without disconnecting all external power sources.

CAUTION
Back up screen programs prior to changing EPROM fimrware. Existing screen programs are clearcd upon power up whenever upgraded or different firmware is installed.

1. Turn off power to the workstation and remove the power cord.

2. Remove the cover plate from the back pancl of the unit by loosening the two screws as shown in Figure 2-3, below. Save all screws and washers.

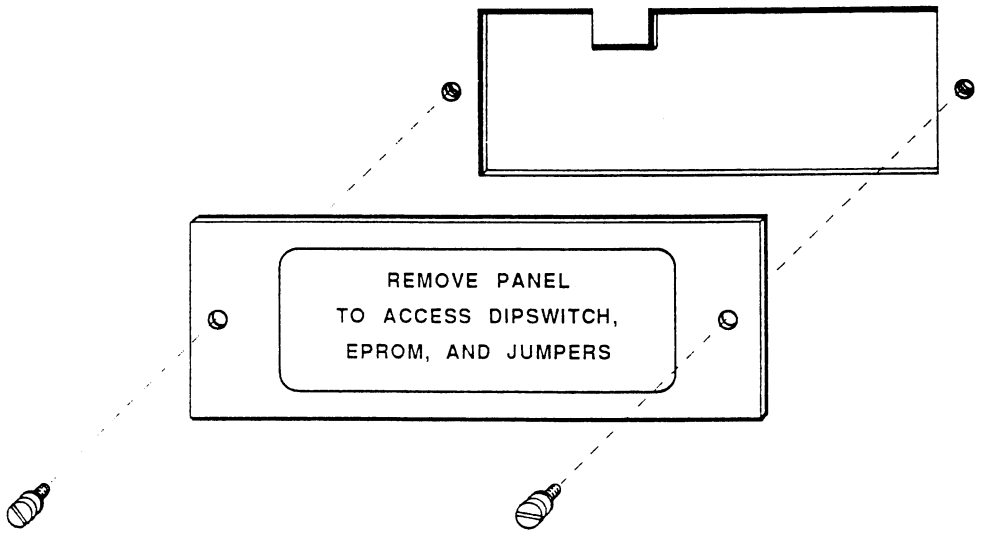


Figure 2-3. Removing the Cover Plate

3. Locate the firmware socket (see Figure 2-1). Remove the 2000-01 firmware chip.

4. Install the replacement firmware chip into the socket, orienting pin one of the chip holder to pin one of the socket, as shown in Figure 2-4, below. Apply gentle, even pressure, and be sure that no pins are bent or out of alignment in the sockets.

5. Replace the cover plate. (If jumpers or switches need to be changed, leave the plate off).

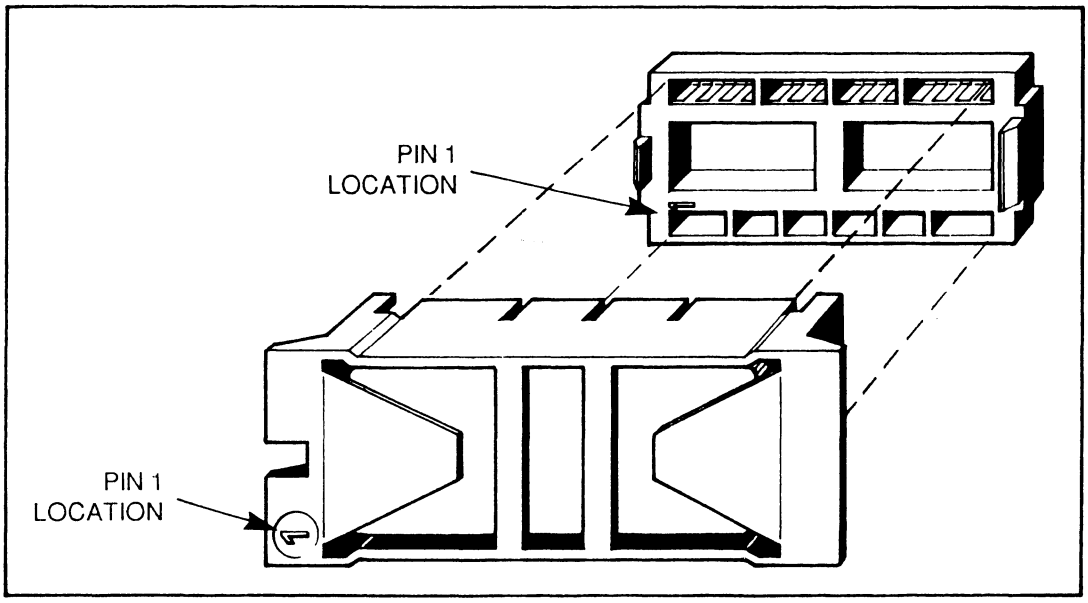


Figure 2-4. Installing the Firmware Chip

2.6.2 Hardware Installation

> **WARNING**
> Before connecting electrical power to the workstation, ensure that the power switch is set to **OFF**. The power cable must be connected to a properly grounded outlet. **DO NOT** use an adapter plug that prevents the workstation from being properly grounded through the power cable.

1. Remove the cover plate from the back panel of the unit by loosening the two screws as shown in Figure 2-3. Save all screws and washers.

2. Install the firmware chip, if necessary (see Section 2.6.1 for instructions).

3. Set the jumpers to configure the serial ports as RS-232 or RS-485. Jumper settings are listed in Section 2.9.

4. Set the switches to define keyboard type, keypad type (for 2000 version of the workstation), and to indicate whether the system has a touch screen. Switch settings are listed in Section 2.10.

5. Replace the cover plate.

6. Secure the 2000 in a suitable mounting location (refer to Appendix A).

7. With a screwdriver, remove the plastic end cap to the fuse receptacle. Check the fuse for serviceability and correct rating (1.5 Amp, 250 V Slo-Blo). Replace cap.

8. Connect any host devices or peripherals to the appropriate ports. See Section 2.11 for pinout descriptions.

9. For 2050, set voltage selector switch to select either 115 VAC or 230 VAC line voltage.

10. Plug the power cable into the power receptacle, as shown in Figure 2.5 on the next page. (See Section 2.6.4 for instructions on creating a power cable.)

11. Plug the end of the power cable into a properly grounded outlet.

12. Set the power switch to **ON**.

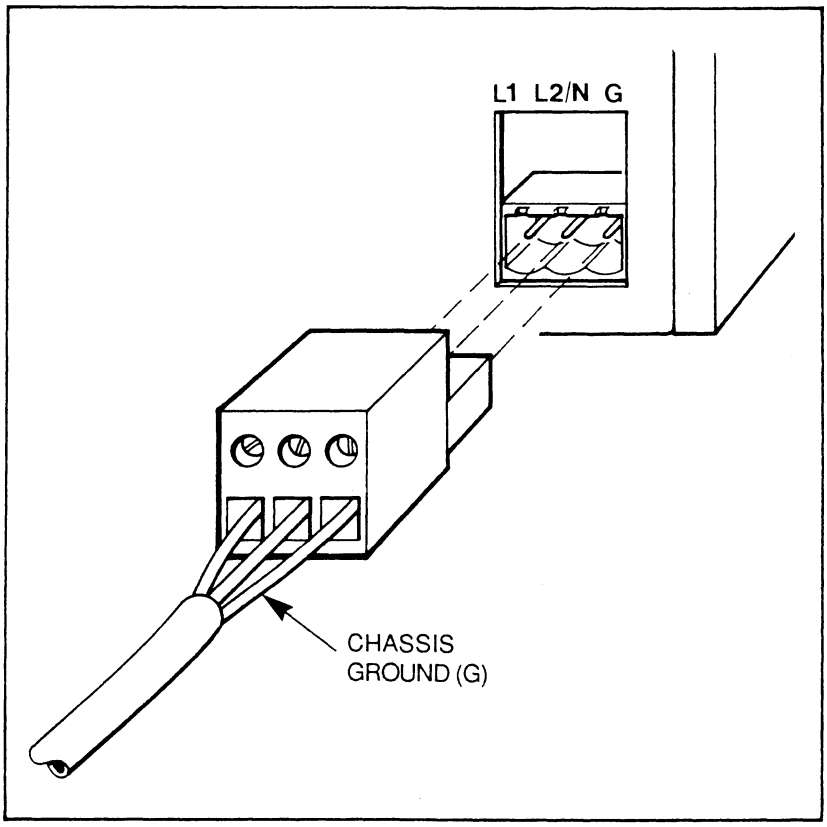


Figure 2-5. Connecting the Power Cable to the Workstation

2.6.3 Creating a Power Cable

A power cable must be created to supply power to the 2000. Materials needed are:

- 3-position terminal block plug (supplied)
- 14, 16, or 18 gauge solid or stranded wire

> **NOTE**
> If stranded wire is used, wire ends should be tinned with solder to keep the wire from fraying.

1. Cut wire cable to the desired length.

2. Strip 1/4 - 3/8 inch of insulation from the other end of the cable. No bare wire should be exposed when the cable is connected to the workstation.

3. Tin the wire ends with solder if using stranded wire. This will keep the wire from fraying.

> **WARNING**
> When inserting the wire ends of the power cable into the block plug, be sure that no bare wire is exposed. Trim the wire ends of the cable or cut a new cable if necessary.

4. Insert the three wire ends of the power cable into the three holes of the block plug, as shown in Figure 2-5. The ground, L1 and L2/N wires should be inserted into the corresponding holes, as indicated in Figure 2-5 also. Be sure that no bare wire is exposed.

5. Tighten the three screws above the wires to hold them firmly in place.

> **WARNING**
> Never tighten the three screws of the block plug when the cable is connected to a power source. The screws are conductive and have full contact with the cable wire.

The power cable is now ready for connecting to the 2000 and an appropriate power source.

> **NOTE**
> The 2000, 2005, and 2060 Workstations have an auto-sensing power supply that automatically adjusts to the voltage supplied (90 - 250 VAC). There are no switches or voltage cards to change. The 2050 has a voltage selector switch of 115 VAC or 230 VAC.

2.7 VERIFYING INSTALLATION

Once the power-up diagnostics are complete, a blank screen will appear. To bring up the Main Menu, press the **<F10>** key of the keyboard twice.

The Main Menu should now appear, as shown in Figure 2-6, below.

> **NOTE**
> The Main Menu shown below is for the base terminal only. If 2000-04 or higher firmware is installed, a different Main Menu will appear. Refer to the 2000 OIL Manual for a description of the OIL menu structure.

```
--Xycom 2000 Base Terminal--
Release X.X

1) Configuration
2) Diagnostics
3) Set Password
4) Set Tab Stops
5) Stored Screen Utilities

<ESC> or <ENTER> to quit
```

Figure 2-6. Main Menu

Each menu option is discussed in full in Chapter 3, Basic Concepts.

2.8 POWER-UP or RESET

When the terminal is powered-up it goes through a set sequence, which consists of:

- Clearing all command and data queues.
- Performing a diagnostic RAM and ROM test.

When the terminal is reset through a remote command, all command and data queues are cleared, but diagnostics are not run.

2.9 SERIAL PORT JUMPERS

Two clusters of jumpers are located on the back panel of the workstation, beneath the cover plate. These jumpers are used to configure the serial ports as either RS-232C (shipping configuration) or RS-485. Jumpers J1 - J12 configure the secondary serial port; jumpers J13 - J24 configure the primary serial port. Figure 2-7, below, shows the arrangement of these jumpers.

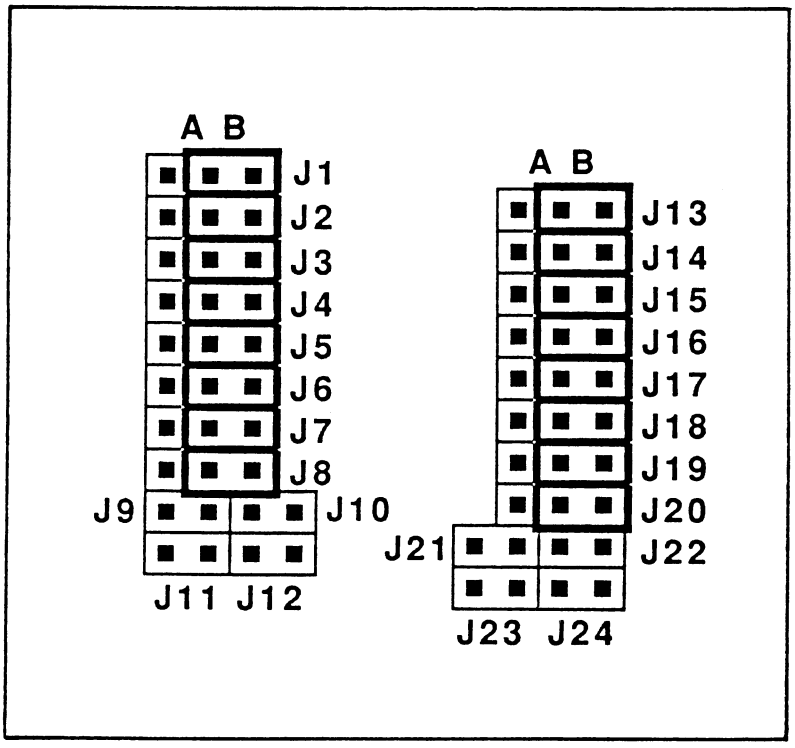


Figure 2-7. Port Configuration Jumpers Set for RS-232

The primary and secondary serial ports may be configured as RS-485 or RS-232C (shipping configuration). Jumper settings are listed in Tables 2-1 and 2-2, below.

Table 2-2. Secondary Serial Port Jumper Settings

Jumper	RS-485/ Multidrop	RS-232*	RS-485
J1	A	B	B
J2	A	B	A
J3	A	B	A
J4	A	B	A
J5	A	B	A
J6	A	B	A
J7	A	B	A
J8	A	B	A

* shipping configuration

Table 2-3. Primary Serial Port Jumper Settings

Jumper	RS-485	RS-232*
J13	A	B
J14	A	B
J15	A	B
J16	A	B
J17	A	B
J18	A	B
J19	A	B
J20	A	B

When configured for RS-485, the inputs CTS and RXD may be terminated. Each signal for each port is independently terminated by a pair of jumpers. Installing the jumpers as indicated below will terminate a specific signal.

Secondary Serial Port	**Primary Serial Port**
RXD: J11, J12 IN	RXD: J23, J24 IN
CTS: J9, J10 IN	CTS: J21, J22 IN

The shipping configuration of these jumpers is all OUT.

2.10 SYSTEM CONFIGURATION SWITCHES

A bank of 8 DIP switches is located on the back panel of the 2000, beneath the cover plate. These switches indicate which type of keypad, keyboard, and matrix keyboard is connected to the system. They also determine where OIL programs will be located (when OIL firmware is installed) and indicate whether the touch screen option is present. Figure 2-9 shows a diagram of the switch bank. When the switch is up (top of the terminal orientation) it is ON. The shipping configuration for the switches is all OFF, unless a touch screen is installed, in which case switch 2 would be ON, and the other switches would be OFF. Figure 2-8 , on the following page, displays the system configuration switches. Switch functions are as follows:

Switch - Keyboard Type	ON = XT keyboard OFF = AT keyboard
Switch 2 - Touch Screen Option	ON = touch screen OFF = no touch screen
Switch 3 - Location of OIL Program	ON = OIL programs in EPROM OFF = OIL programs in RAM
Switches 4 and 5 - Matrix Keyboard Type	Both OFF = QWERTY type 4 ON, 5 OFF = ABC type
Switches 6 and 7 - Keypad Type	Both OFF = type 0 2005 6 ON, 7 OFF = type 1 Reserved 6 OFF, 7 ON = type 2 2050/2060 BOTH ON = type 3 Reserved
Switch 8 - Reserved	

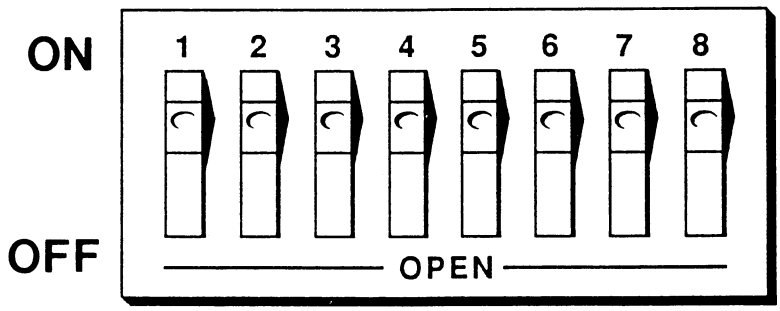


Figure 2-8. System Configuration Switches

2.11 PORTS

2.11.1 Primary and Secondary Serial Ports

There are two asynchronous serial ports on the back panel of the 2000. The primary serial port is located beneath the secondary serial port (see Figure 2-1). Both ports are male, 9-pin D-type subminiature connectors. Both ports are configurable, via jumpers, as either RS-232C or RS-485 (see Section 2.9 for jumper settings). Pinouts for each configuration are listed in Table 2-4, and Table 2-5 on the following pages.

Table 2-4. Primary Serial Port Pinouts

Pin #	RS-232	RS-485
1	DCD	TXD-
2	RXD	TXD+
3	TXD	RTS-
4	DTR	RTS+
5	GND	GND
6	NC	RXD-
7	RTS	RXD+
8	CTS	CTS+
9	NC	CTS-

Table 2-5. Secondary Serial Port Pinouts

Pin #	RS-232	RS-485
1	NC	TXD-
2	RXD	TXD+
3	TXD	RTS-
4	NC	RTS+
5	GND	GND
6	NC	RXD-
7	RTS	RXD+
8	CTS	CTS+
9	NC	CTS-

2.11.2 **Matrix Parallel Keyboard Port**

Two matrix-type keyboards are available for use with the 2000 Industrial Workstation: a QWERTY-type sealed membrane keyboard with 20 function keys, and an ABC-type 58-key sealed membrane keypad, also with 20 function keys. These keyboards require a special 26-pin male parallel connection. This connection is available on the back panel of the 2000 (see Figure 2-1 for port location). The pinouts for the matrix parallel keyboard port are shown in Table 2-6, below.

Table 2-6. Matrix Parallel Keyboard Port Pinouts

Pin #	Function	Pin #	Function
1	Column B	14	Row 7
2	Row 1	15	Column 5
3	Column 11	16	Row 8
4	Row 2	17	Column 7
5	Column 10	18	Column 12
6	Row 3	19	Column 6
7	Column 3	20	Column 13
8	Row 4	21	Column 0
9	Column 2	22	No Connect
10	Row 5	23	Numlock Rstr.
11	Column 1	24	Numlock
12	Row 6	25	Caplock Rstr.
13	Column 4	26	Caplock

2.11.3 **Parallel Input/Output Port**

2.11.3.1 **Output**

A 25-pin D-type female parallel port is available on the back panel of the 2000 as a Centronics compatible output port. This port may also be used as an input port (see Section 2.11.3.2). Pinouts for this port are given in Table 2-7.

Table 2-7. Parallel Input/Output Port Pinouts

Pin #	Signal	Pin #	Signal
*1	STROBE	10	ACK
*2	DATA0	11	BUSY
*3	DATA1	12	NC
*4	DATA2	13	NC
*5	DATA3	14	NC
*6	DATA4	15	NC
*7	DATA5	16	RESET
*8	DATA6	17	NC
*9	DATA7	*18-25	GND

* Used for input mode
All signals are used for printer mode

2.11.3.2 **Input**

If serial printing is selected via the Miscellaneous Configuration Menu (see Section 3.4.3), the parallel printer port is configured as a parallel input port. In this configuration commands can be issued and programs executed in a parallel fashion over the port.

When configured as a parallel input port, 8 bits of data are presented on the data input lines. After the prescribed data setup time a strobe is issued. Data must be held for the prescribed data hold time (before changing to the next set of data). The edge of the strobe pulse responsible for latching the data is selectable via the Miscellaneous Configuration Menu (described in Section 3.4.3). When High True Strobe is selected, the STB input pin is normally low. Raising the STB input from low to high clocks in the data and causes an interrupt within the terminal. When Low True Strobe is selected, lowering the STB input from high to low causes the data to be clocked in. The timing requirements for the two strobe polarities are shown in Figures 2.10 and 2.11 on the next page.

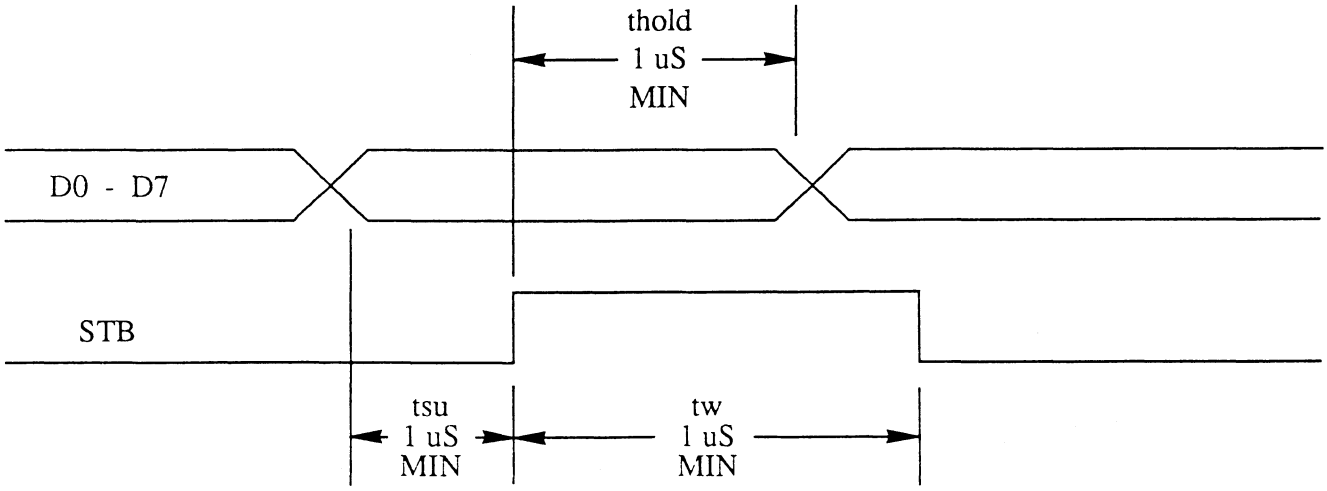


Figure 2-9. High True Strobe Timing

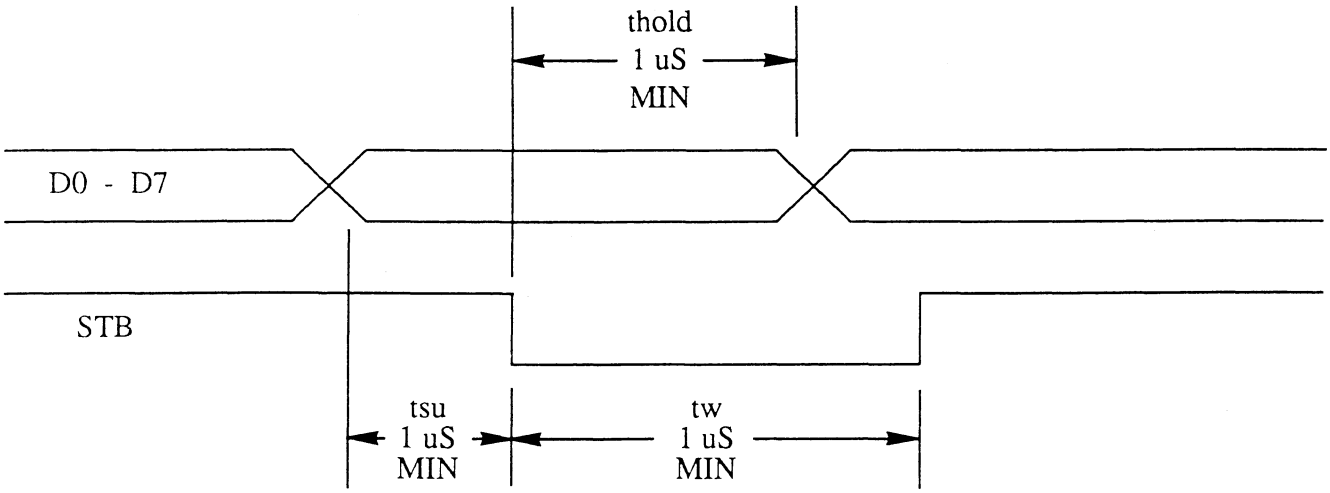


Figure 2-10. Low True Strobe Timing

2.11.4 **Keyboard Port**

A standard 5-pin keyboard connector is available on the back panel of the 2000. See Figure 2-1 for location of this connector. Pinouts for this connector are listed in Table 2-8.

Table 2-8. Keyboard Connector Pinouts

Pin #	Signal
1	Clock
2	Data
3	NC
4	GND(SG)
5	+5 VDC
6	GND(FG)

2.12 REPLACING THE BACKUP BATTERY

The backup battery is located under the round slotted cap in the center of the upper back panel of the unit. To remove it, turn the cap with a screwdriver counter-clockwise, as shown in Figure 2-11. Pry the cap off and remove the battery. Replace with a new battery. Due to an internal capacitor, the battery can be removed for up to 10 minutes without losing data. If the unit is off prior to battery exchange, turn it on for 10 seconds and then back off to recharge the internal capacitor.

The typical battery life is 10 years at 25° C; worst case is 3 years at the non-operating storage temperature of 65° C. For routine maintenance replace the battery every 3 years in order to guarantee date integrity of screen programs stored in battery-backed CMOS RAM.

CAUTION

DANGER OF EXPLOSION IF BATTERY IS INCORRECTLY REPLACED. Replace only with the same or equivalent type recommended by the manufacturer. Dispose of used batteries according to the manufacturer's instructions.

Bei falschem Umgang mit oder falschem Einbau einer Lithium-Batterie kann eine Explosion entstehen, bel der in der Nahe befindliche Personen schwere Verietzungen erieiden konnen. Versuchen Sie nicht, Lithium-Batterien wieder aufzuiaden, kurzzuschliessen oder zu ottnen, und werfen Sie sie nicht in den Mull oder in ein Feuer. Wechsein Sie sie nur gegen gensu den gleichen Typ aus. Zur Entsorgung mussan Sie Lithium-Batterien an ihren Handler zuruckgeben.

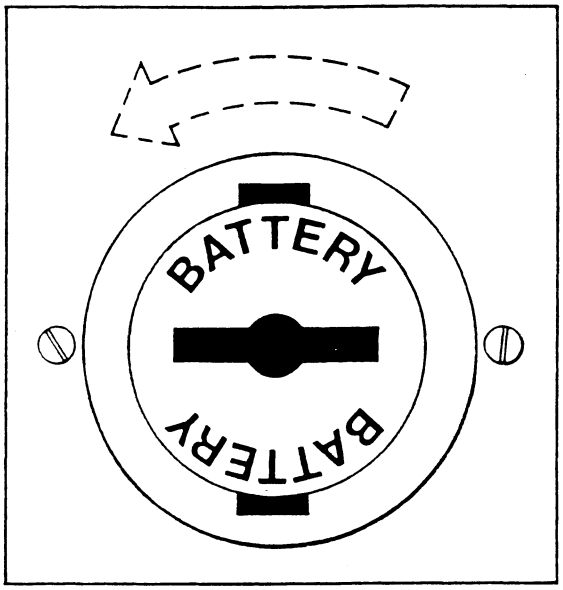


Figure 2-11. Removing the Battery Cap

2.13 ADJUSTING VIDEO CONTRAST

The 2000 Series Industrial Workstations are adjusted for proper video contrast prior to shipping. If video contrast requires adjusting, it can be done via an adjustment mechanism located on the bottom panel of the 2000/2005 units, the back panel (finger adjustment) for the 2050 unit, and the side panel for the 2060 unit. This 1/4 inch hole, labeled "VIDEO CONTRAST," is located in the center of the far left side of the bottom panel (front panel down orientation). To adjust the video contrast, simply insert a small screwdriver in the hole and gently twist until the desired contrast is attained. However, this mechanism is factory adjusted and should need no further adjustment.

Similar adjustment mechanisms are located on the right-side panel of the unit (facing the front panel orientation). These clearly labeled mechanisms are for adjusting focus, brightness, width, horizontal center, vertical linearity, vertical hold, and vertical size. Although these display qualities may also be adjusted with a small screwdriver, such adjustments should not be necessary, and are not recommended. Please refer all problems concerning video display adjustment to the Xycom Application Engineering Department.

3.1 INTRODUCTION

This chapter introduces the basic programming and application concepts of the Xycom 2000 Industrial Workstation with the 2001 (base terminal) firmware option.

3.2 MENU HIERARCHY

The base terminal firmware allows the 2000 Workstation to operate in two modes: Operating Mode and Set-Up Mode. When in Operating Mode the workstation can execute program blocks and process remote commands. When in Set-Up Mode, the 2000 displays menus on the screen which allow the user to change its configuration, type programs into program blocks, or edit existing programs.

Upon power-up, the 2000 is automatically in Operating Mode. To enter Set-Up Mode, do either of the following:

- Press the <F10> key twice on the keyboard
  or
- Press upper right and lower right keys simultaneously on the keypad, which are PG DN and 9 on the 2005 keypad and C and ENTER for the 2050/2060 keypad

The Main Menu should appear on the screen.

If the password has been enabled the user will be prompted to type a 3-letter password. If the password is correctly entered the Main Menu will be displayed on the screen (see Section 3.6 for password information).

In order for the workstation to receive and execute commands from the control system, it must be returned to Operating Mode. To return to Operating Mode from the Main Menu, simply press the <Esc> or <Enter> key once. If you are in any of the secondary menus you will have to press <Esc> or <Enter> once to return to the Main Menu. Then press the key again to go to Operating Mode.

3.3 MAIN MENU

When the 2000 enters Set-Up Mode, the Main Menu will appear, as shown in Figure 3-1 below, with the following options:

> **NOTE**
> If you have firmware 2000-06 or higher, see the 2000-OIL manual.

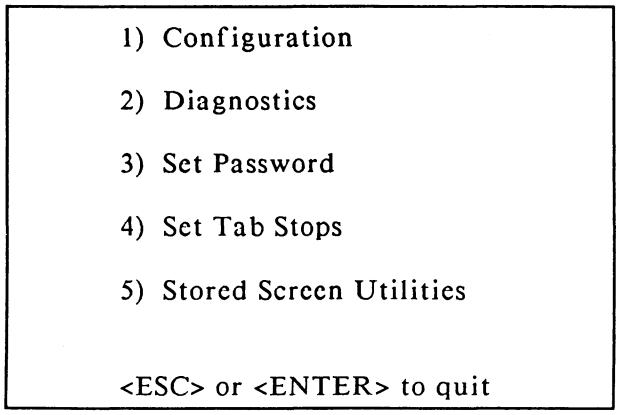


Figure 3-1. Main Menu

Selecting any of the options (option number) will bring up secondary menus with additional options. Figure 3-2 shows the tree structure of the 2000 Workstation option menus and the relationship of each option when it is in Set-Up Mode.

The following sections provide an overview of each option from the Set-Up Mode Main Menu.

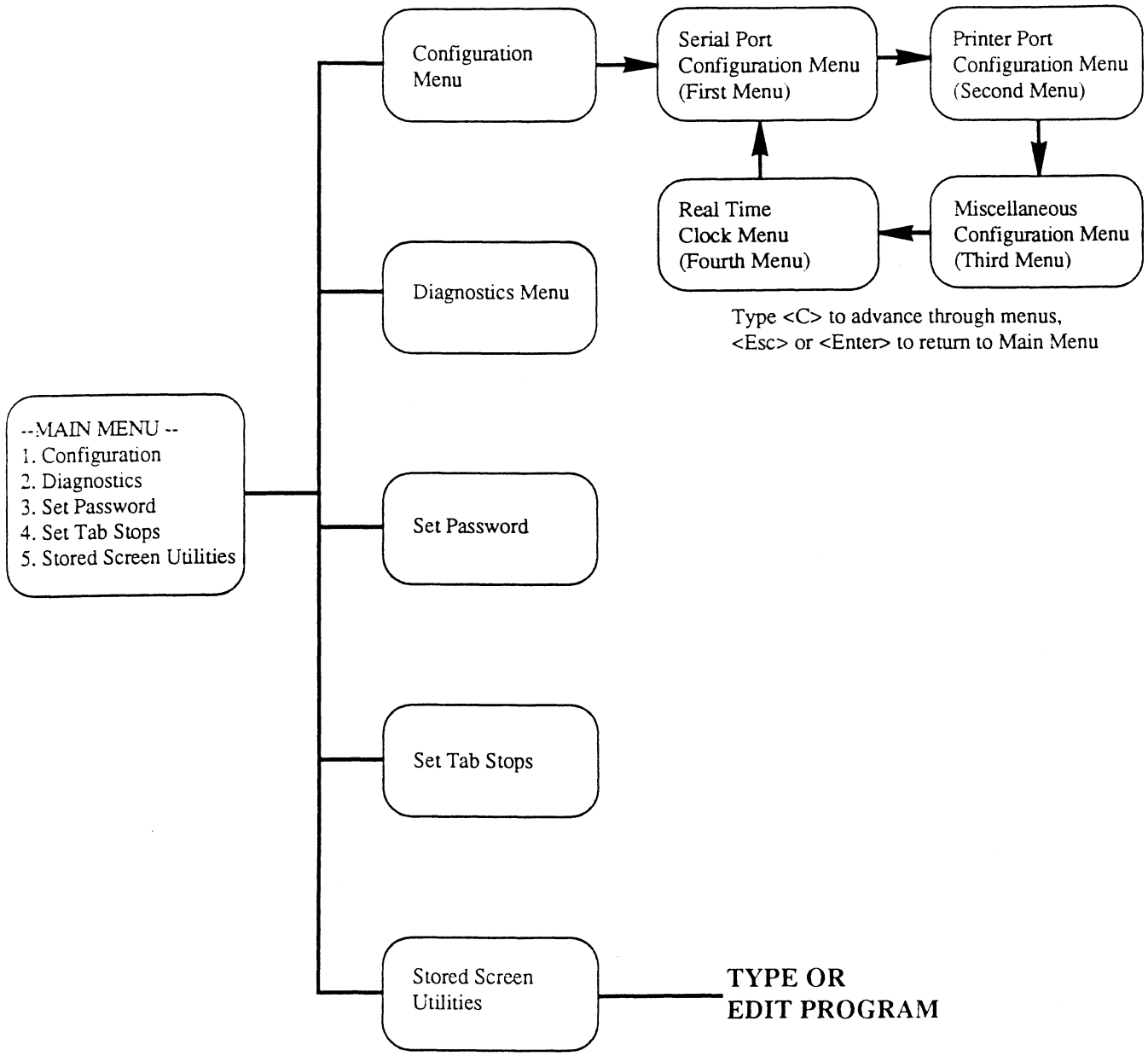


Figure 3-2. Menu Hierarchy

3.4 CONFIGURATION MENUS

There are four configuration menus. When you press "1" while in the Main Menu, the first configuration menu is displayed on the screen. To get to any subsequent menus, press "C" while in the current menu.

3.4.1 Serial Port Configuration Menu (First Configuration Menu)

The first configuration menu, titled Serial Port Configuration, (see Figure 3-3), comes up when the Configuration Menu option is entered. This is the configuration for the primary serial port. If "C" is typed from this menu, the second menu is displayed.

```
                    -- Serial Port Configuration Menu --

6   Baud - 1=300 2=600 3=1200 4=2400 5=4800 6=9600 7=19.2K
0   Parity - 0=Zero 1=One 2=Even 3=Odd
0   1=Parity Enabled            0=Disabled
1   1=8 Data Bits               0=7 Data Bits
1   1=Full Duplex               0=Half Duplex
0   1=Handshaking Enabled       0=Disabled
0   1=RTS/CTS Handshaking       0=Xon/Xoff Handshaking
0   1=RS-485                    0=RS-232
0   0=Xon/Xoff Receive - 1=Xon/Xoff Transmit - 2=Xon/Xoff Transmit/Receive


    Use <UP-ARROW>, <DOWN-ARROW>, <LEFT-ARROW>, <RIGHT-ARROW>
    Use values 0 through 9
    "C" for next configuration menu, <ESC> or <ENTER> to quit
```

Figure 3-3. Serial Port Configuration Menu

The first column of the menu lists the current settings of all the configuration options. The available options and their corresponding settings are listed to the right.

To change a configuration option, first move the cursor to the row containing the value which is to be changed. This is done by pressing the up-arrow and down-arrow keys. When the cursor is properly positioned, press a number to select the desired option. After all changes are made, press <Esc> or <Enter>.

Baud Rate. The baud rate of the serial channel should be set to match the serial device connected to the workstation.

Parity. If parity is enabled (see the next item), the type of parity used by the workstation must be set to match that used by the serial device. The types of parity that can be selected are:

Primary Port

0 = parity bit always 0
1 = parity bit always 1
2 = even parity
3 = odd parity

Disable or Enable Parity. Parity bit insertion and checking can be enabled or disabled. Parity should be enabled if the connected device uses parity bit insertion and checking; otherwise, parity should be disabled. If parity is enabled, the type of parity must also be selected (see previous item).

> **NOTE**
> Disable parity and seven data bits cannot be selected simultaneously.

Data Bits. Number of data bits per byte (7 or 8) should be set to match the serial device.

> **NOTE**
> Seven data bits and parity disabled cannot be selected simultaneously. If the terminal is configured for 7 bits and no parity, it will use 8 bits and no parity instead.

Full/Half Duplex. If the connected device is capable of simultaneous two-way communications and is set up for echoing, the workstation should be used in full-duplex mode. If echoing is not used or the host is not capable of simultaneous two-way communications, select half-duplex mode.

> **NOTE**
> When the unit is configured for half-duplex, the RTS line takes on a special function.

When a character is transmitted from the workstation, RTS will go high and remain high until one of the terminating characters are transmitted:

<CR>	Carriage Return ASCII 13 (decimal)
<ETX>	End of Text ASCII 3 (decimal)
<EOT>	End of Transmission ASCII 4 (decimal)

When the termination character is transmitted, RTS will go low and remain there until the next non-termination character is transmitted.

Handshaking Enabled. Must be set to 1 to enable either RTS/CTS or XON/XOFF handshaking. If handshaking is enabled with half-duplex selected, the workstation will ignore handshaking and disable it when you leave the configuration menu. In addition, if full-duplex is selected and handshaking disabled, RTS will always be high.

> **NOTE**
> Handshaking and half-duplex cannot be enabled simultaneously.

RTS/CTS Handshaking, XON/XOFF.

1 =	RTS/CTS handshaking
0 =	XON/XOFF generation

RTS/CTS Handshaking enabled. Handshaking is accomplished through hardware in the following manner:

RTS is an output from the workstation. It will be asserted (High) when it is time for an external device to send data to the workstation. When RTS is inactive (Low), the sending unit should not attempt to send data. This protects the workstation from input buffer overflow.

CTS is an input to the workstation. If this line is asserted (High) the workstation assumes that it is time to transmit data to an external device. When CTS is inactive (Low) the workstation

will stop transmitting data to an external device. This keeps the workstation from overflowing the input buffers on an external device.

XON/XOFF Handshaking enabled. Handshaking is accomplished through software in the following manner:

An XON (DC1 ASCII 17 decimal) will be sent by the workstation when it is time for the external device to send data.

If an XON is received by the workstation, it will assume that it is time to send data to an external device.

An XOFF (DC3 ASCII 19 decimal) will be sent by the workstation when the sending device should stop sending data.

If an XOFF is received by the workstation, it will stop sending data until an XON is received.

> **NOTE**
> Care should be taken when using XON/OFF handshaking. If the data stream being transmitted contains the XOFF (ASCII 19 or DC3) character, an operator could inadvertently disable communications.

RS-485/RS-232. This option selects whether the port is transmitting RS-232 or RS-485 signals. It informs the workstation firmware of serial port jumper settings. For proper serial port jumper settings, see Section 2.9.

XON/XOFF Transmit/Receive. This option selects whether XON/XOFF will be only transmitted by the workstation, only received by the workstation, or if XON/OFF will be both transmitted and received.

If set to Transmit, the workstation will transmit but not receive XON/XOFF.

If set to Receive, the workstation will receive but not transmit XON/XOFF.

If set to Transmit/Receive, the workstation will both transmit and receive XON/XOFF.

3.4.2 Printer Port Configuration Menu (Second Configuration Menu)

The second configuration menu, Printer Port Configuration, as shown in Figure 3-4, is displayed if "C" is typed from the first configuration menu. This menu configures the secondary serial port, **not** the parallel printer port. (From this menu, typing another "C" brings up the third configuration menu.)

```
                    --Printer Port Configuration Menu--

6  Baud - 1=300 2=600 3=1200 4=2400 5=4800 6=9600 7=19.2K
0  1=Parity Enabled          0=Disabled
0  1=Odd Parity              0=Even
1  1=8 Data Bits             0=7 Data Bits
0  1=Handshaking Enabled     0=Disabled
0  1=RTS/CTS Handshaking     0=Xon/Xoff Handshaking


   Use <UP-ARROW>, <DOWN-ARROW>, <LEFT-ARROW>, <RIGHT-ARROW>
   Use values 0 through 9
   "C" for next configuration menu, <ESC> or <ENTER> to quit
```

Figure 3-4. Printer Port Configuration Menu

Baud Rate. The baud rate of the printer port should be set to match the baud rate of the printer connected to the workstation.

Disable or Enable Parity. Parity bit insertion and checking can be enabled or disabled. Parity should be enabled if the connected device uses parity bit insertion and checking; otherwise, parity should be disabled. If parity is enabled, the type of parity must also be selected (see next item).

> **NOTE**
> Disable parity and seven data bits cannot be selected simultaneously.

Parity. If parity is enabled, the type of parity used by the workstation must be set to match that used by the printer. The types of parity that can be selected are:

Printer Port

0 = even parity
1 = odd parity

Data Bits. Number of data bits per byte (7 or 8) should be set to match the serial device.

> **NOTE**
> Seven data bits and parity disabled cannot be selected simultaneously. If the terminal is configured for 7 bits and no parity, it will use 8 bits and no parity instead.

Handshaking Enabled. Must be set to 1 to enable either RTS/CTS or XON/XOFF handshaking.

RTS/CTS Handshaking, XON/XOFF.

1 = RTS/CTS handshaking
0 = XON/XOFF generation

RTS/CTS Handshaking enabled. Handshaking is accomplished through hardware in the following manner:

RTS is an output from the workstation. It will be asserted (High) when it is time for an external device to send data to the workstation. When RTS is inactive (Low), the sending unit should not attempt to send data. This protects the workstation from input buffer overflow.

CTS is an input to the workstation. If this line is asserted (High) the workstation assumes that it is time to transmit data to an external device. When CTS is inactive (Low) the workstation will stop transmitting data to an external device. This keeps the workstation from overflowing the input buffers on an external device.

XON/XOFF Handshaking enabled. Handshaking is accomplished through software in the following manner:

An XON (DC1 ASCII 17 decimal) will be sent by the workstation when it is time for the external device to send data.

If an XON is received by the workstation, it will assume that it is time to send data to an external device.

An XOFF (DC3 ASCII 19 decimal) will be sent by the workstation when the sending device should stop sending data.

If an XOFF is received by the workstation, it will stop sending data until an XON is received.

> **NOTE**
> Care should be taken when using XON/OFF handshaking. If the data stream being transmitted contains the XOFF (ASCII 19 or DC3) character, an operator could inadvertently disable communications.

3.4.3 Miscellaneous Configuration Menu (Third Configuration Menu)

The third configuration menu, Miscellaneous Configuration, as shown in Figure 3-5, is displayed if "C" is typed from the second configuration menu. (From this menu, typing another "C" brings up the fourth configuration menu.)

```
                    -- Miscellaneous Configuration Menu --

 0          1=ANSI Emulation                  0=Hazeltine 1500
 1          1=Display Control Characters      0=Normal Display
 1          1=Enable Auto Line Feed           0=Disable
 1          1=Enable Autowrap                 0=Disable
 1          1=Enable Scrolling                0=Disable
 0          1=Block Cursor                    0=Underline Cursor
 1          1=60 Hz.                          0=50 Hz.
 0          1=Lock Menu Entry From Keypad     0=Unlock
 1          1=Parallel Printer                0=Serial Printer
 0          Parallel Input Port Strobe:       0=Low True  1=High True
01          Number of Status Lines (0 - 24)


Use <UP-ARROW>, <DOWN-ARROW>, <LEFT-ARROW>, <RIGHT-ARROW>.
Use values 0 through 9.
"C" for next configuration menu, <ESC> or <ENTER> to quit
```

Figure 3-5. Miscellaneous Configuration Menu

ANSI Emulation/Hazeltine 1500. The 2000 can emulate either a Hazeltine 1500 terminal or an ANSI x3.64 terminal. Hazeltine 1500 and ANSI emulation differ in the character sequences which must be transmitted to the terminal to execute a remote command. For example, to perform the remote command Cursor On, a terminal configured as a Hazeltine 1500 must be sent the character sequence 7EH 02H (or the ASCII characters ~<STX>). However, if configured as an ANSI terminal, the same command requires the character sequence <ESC> [= 1 h (ASCII). Chapter 6 lists both the Hazeltine 1500 and ANSI character sequences which must be transmitted to the workstation to perform remote commands.

Some keys return different characters when the workstation is operating in ANSI mode than when the workstation is operating in Hazeltine mode. See Chapter 4 for a complete list of characters generated by keys in both modes.

ANSI emulation provides support for most DEC VT100/220 remote commands. The VT100/220 commands which are supported are listed in Chapter 6, while VT100/220 commands not supported are listed in Appendix C.

Display Control Characters/Normal Display. During normal operation, the terminal executes control characters that it receives such as carriage return, linefeed, etc. In addition to this mode, the terminal can be made to simply display control codes and not execute them. When the terminal displays a control code, it shows a two-letter abbreviation of the ASCII control code (see Table 3-1) in a single character space. Displaying control codes is useful when installing and testing communications.

Table 3-1. Two-letter Abbreviations of ASCII Control Codes

Hexadecimal Code	ASCII Code	Two-letter Abbreviation
00	NUL	NL
01	SOH	SH
02	STX	SX
03	ETX	EX
04	EOT	ET
05	ENQ	EQ
06	ACK	AK
07	BEL	BL
08	BS	BS
09	HT	HT
0A	LF	LF
0B	VT	VT
0C	FF	FF
0D	CR	CR
0E	SO	SO
0F	SI	SI
10	DLE	DL
11	DC1 (XON)	D1
12	DC2	D2
13	DC3 (XOFF)	D3
14	DC4	D4
15	NAK	NK
16	SYN	SY
17	ETB	EB
18	CAN	CN
19	EM	EM
1A	SUB	SB
1B	ESC	EC
1C	FS	FS
1D	GS	GS
1E	RS	RS
1F	US	US

Enable Automatic Linefeed/Disable. If automatic linefeed is enabled, the cursor will automatically perform a linefeed after it receives and executes a carriage return. Linefeeds are ignored. If disabled, only a carriage return will be executed when a carriage return is received (linefeeds are executed as linefeeds).

Enable Autowrap/Disable. If autowrap is enabled, lines more than 80 characters long will wrap around to the next line. If disabled, any character issued after column 80 will be printed in column 80. Autowrap is automatically enabled if Hazeltine 1500 emulation is selected.

Enable Scrolling/Disable. If scrolling is disabled, moving the cursor below the last line in the screen will cause the cursor to wrap to the top of the screen.

Block Cursor/Underline Cursor. Either an underline or a block cursor can be chosen. Both types of cursors are blinking. The block cursor is more visible than the underline cursor.

60/50 Hz. This option should be set to match the frequency of the AC power source: usually 60 Hz in the United States, 50 Hz in Europe. This option has no affect in the 2050.

Lock Menu Entry From Keypad/Unlock. If menu entry is unlocked, the menus can also be entered from the keypad and sealed keyboard. If the keypad is locked out, the password prompt can only be invoked by pressing "F10" twice on the full-stroke keyboard. This prevents entry to the Set-up Mode from the keypad.

Parallel/Serial Printer. If parallel printer is selected, printing will occur through the Centronics parallel port. If serial printer is selected, printing will occur through the primary serial port. When serial printing is selected, the parallel port will accept data input.

> **NOTE**
> To use the parallel port for data input, printing must occur through the secondary serial port.

Parallel Input Port Strobe. If the parallel port is being used for input, this menu option selects the voltage level of the strobe. When a program is strobed into the port, it will be executed unconditionally. To use the parallel port for data input, serial printing must be selected. (For a description of strobe timing, see Section 2.11.3.2.)

Number of Status Lines (0-24). If this menu selection is chosen, the user will have from 0 to 24 non-scrolling status lines to be maintained at the bottom of the display. The full area of the screen, including the 25th line, is addressable.

The (0,0) coordinate for the PLOT and UNPLOT commands is the lower left corner of the scrolled area.

3.4.4 Real Time Clock Configuration Menu (Fourth Configuration Menu)

The fourth configuration menu, Real Time Clock Configuration, as shown in Figure 3-6, is displayed if "C" is typed from the third configuration menu. To return to the first configuration menu, type "C" at this menu.

```
                    --Real Time Clock Configuration Menu--

  0          1=Clock Display Enabled    0=Clock Display Disabled
  0          1=24 Hour Clock            0=12 Hour Clock
 00          Year
 01          Month
 01          Day
 00          Hour
 00          Minute
 00          Second


 Use <UP-ARROW>, <DOWN-ARROW>, <LEFT-ARROW>, <RIGHT-ARROW>
 Use values 0 through 9
 "C" for next configuration menu, <ESC> or <ENTER> to quit
```

Figure 3-6. Real Time Clock Configuration Menu

Clock Display. When the Clock Display is enabled, a date will appear in the lower left corner of the screen, and the time will appear in the lower right corner of the screen while the workstation is in Operating mode and during program execution.

24 Hour Clock/12 Hour Clock. This option selects whether the clock display will use a 24 hour military clock or a 12 hour clock with AM and PM designations. Enter the hour using 0 - 23 even if the 12 hour clock format is slected.

year, month, day, hour, minute, seconds. Whenever the workstation is powered up after the CMOS RAM has been powered down or the battery removed, the clock/calendar will be initialized to 01/01/00, 00:00:00.

Data Registers #1 - #7 are continuously updated with the current year, month, day, hour, minute, second, and day-of-week, respectively. (The year, month, and day are automatically adjusted for leap years for any date from 1950 to 2050.) These seven registers cannot be altered by a program running in the workstation -- they can only be read. Their values can be changed only through the Miscellaneous Configuration Menu or through the "Set Time" remote command.

3.5 DIAGNOSTICS MENU

Selecting option #2 from the Main Menu brings up a Diagnostics Menu for general purpose testing of RAM, ROM, ports, character attributes, the CRT, and the time-of-day clock/calendar. Figure 3-7 shows the options available from the Diagnostics Menu. When a selected test is completed, status information about the completed test and the Diagnostics Menu will be displayed.

> **NOTE**
> If a firmware option is installed, the Diagnostics Menu may be a subset of the menu shown below. See Appendix A for more information.

```
-- Diagnostics --

1) Complete Test
2) Continuous Test
3) RAM
4) ROM Checksum
5) EPROM Test
6) Real Time Clock Test
7) RS-232 Serial Loopback
8) RS-485/Multidrop Serial Loopback
9) Printer Port Test
A) Parallel Input Test
B) Matrix Keyboard Loopback Test
C) Beeper Test
D) Battery Test
E) Dipswitch Test
F) Character Attributes
G) CRT Crosshatch Pattern
H) CRT Brightness Pattern
I) Character Generator Test
J) Touch Screen Test (if touch screen installed and dipswitch ON)

<ESC> or <ENTER> to quit
```

Figure 3-7. Diagnostics Menu

3.5.1 Complete Test

The complete test exercises all the tests listed on the Diagnostics Menu, one at a time. The user is prompted to install serial loopback connector(s) before beginning the tests. After each test is run, the user is prompted to hit any key to continue. See the following sections for descriptions of the specific diagnostic tests.

3.5.2 Continuous Test

> **CAUTION**
> Turning off power while the continuous test is in progress could destroy all clock data and data in CMOS RAM.

In this mode, the workstation continuously cycles through the RAM, serial port, parallel input port, ROM, and real time clock tests. If an error is found, the workstation stops testing and displays an appropriate error message along with the prompt:

Press any key to continue.

If a key is then pressed, testing will continue. Press any key twice to discontinue testing.

After all testing is complete, the program will continue indefinitely. To exit the continuous test mode, press any key.

3.5.3 RAM Test

If the RAM test is selected, the workstation will check the CPU RAM, the CMOS RAM, the display RAM, the attribute RAM, and the character generator RAM. After checking the CPU RAM the workstation will display one of the following messages:

CPU RAM OK
or
CPU RAM failure

CAUTION
Turning off power while the CMOS RAM test is in progress will destroy all data in CMOS RAM.

The next test checks the CMOS RAM. After testing, the workstation displays one of these messages:

CMOS RAM OK
CMOS RAM failure

The workstation will then test the display RAM, during which a pattern will be flashed on the video display followed by one of these messages:

Display RAM OK
or
Display RAM failure Page #nn

The workstation will then test the attribute RAM, again flashing a pattern on the video display followed by one of these messages:

Attribute RAM OK
or
Attribute RAM failure Page #nn

The workstation will then test the character generator RAM. A pattern will flash on the video display, followed by one of these messages:

Character Generator RAM OK
or
Character Generator RAM failure

3.5.4 ROM Checksum

This test "ROM checksum is: nnnn Should be: nnnn" on the status line. The two checksums listed (nnnn) should match.

3.5.5 EEPROM Test

When this test is selected, a non-destructive write/read test of all registers in the EEPROM is performed, and the message:

EEPROM test in process

appears on the screen. When the test is complete, one of the following messages will appear:

EEPROM device OK
or
EEPROM failure Register #nn

3.5.6 Real Time Clock Test

> **CAUTION**
> Turning off power while the real time clock test is in progress will destroy all clock data.

A non-destructive storage test, counter roll-over test, and control signal test is performed on the clock calendar. If no errors are found, the workstation will display the message "Real Time Clock OK." If an error is found, the subtest which failed and error information will be displayed instead.

3.5.7 RS-232 Serial Loop Back Test

This test checks the primary and secondary serial ports for the RS-232 configuration. Before these ports are tested, a serial loopback connector must be installed, and jumpers must be set to configure the ports as RS-232 (see Section 2.9 for jumper settings). The test plugs should be constructed of a DE-9S connector and jumper wires. The configuration of the test plugs is shown in Figure 3.8 on the next page.

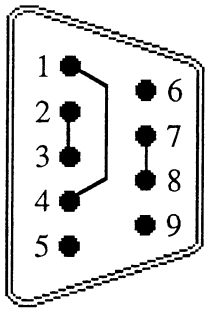


Figure 3-8. Serial Port Test Plugs, RS-232

If the serial ports are operating correctly, the workstation will display the message:

Prim. port: OK. Sec. port: OK.

If an error is found, the workstation will display one of the following messages:

Time out err.
Data err.
CTS-RTS err.
DTR-DSR err.

3.5.8 RS-485/Multidrop Serial Loopback Test

This test checks the primary and secondary serial ports for RS-485 or RS-485/multidrop configuration. Before these ports are tested, serial loopback connectors must be installed, and jumpers must be set to configure the ports as RS-485 or RS-485/multidrop (see Section 2.9 for jumper settings). The test plugs should be constructed of a DE-9S connector and jumper wires. The configuration of the test plugs is shown in Figure 3-9 on the next page.

If the serial ports are operating correctly, the workstation will display the message:

Prim. port: OK. Sec. port: OK.

If an error is found, the workstation will display one of the following messages:

Time out err (RS-485).
Data err (RS-485).
Timeout err (RS-485/multidrop).
Data err (RS-485/multidrop).

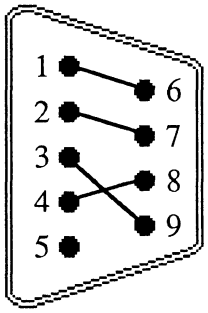


Figure 3-9. Serial Port Test Plugs, RS-485

3.5.9 **Printer Port Test**

To run this test, a printer cable must be attached to the parallel port at the rear of the unit, and the printer must be on-line.

The test pattern that is sent to the printer should be the same as the message that appears on-screen. If the printer port is operating correctly, the workstation will display the message:

Printer Port test passed.

If an error is found, the workstation will display the message:

Printer Error: <ESC> to Abort, any other key to continue

3.5.10 **Parallel Input Port Test**

This test checks the parallel port for input capability. If the port passes the diagnostics, the workstation will display the message:

Parallel Input Port test passed.

If an error is found, the workstation will display the message:

Parallel Input Port test failed.

3.5.11 **Matrix Keyboard Loopback Test**

This test checks the matrix keyboard port. Before the port is tested, a column driver loopback connector must be installed.

If the matrix keyboard port is operating correctly, the workstation will display the message:

Matrix Test Passed. - Remove Loopback.

If an error is found, the workstation will display the following message:

Matrix Test Failed. - Remove Loopback.

3.5.12 **Beeper Test**

This test checks the beeper. If the beeper is functioning, the workstation will emit a continuous beep, and the following message will appear on the screen:

----- Beeper should be sounding. Press any key to exit test. -----

> **NOTE**
> Beeper will not beep if Beeper Disabled Jumper is pulled.

3.5.13 **Battery Test**

This test checks current condition of the battery. If the battery is functioning properly, the screen will display the message:

Battery Test Passed.

If an error is found, the screen will display the message:

Battery Test Failed.

3.5.14 **Dipswitch Test**

To test the dipswitches, bring up the dipswitch test screen and toggle the switches. Closed settings are represented by "C"; open settings are represented by "O." When all the switches are closed (on), the display should look like this:

```
Switch:        8 7 6 5 4 3 2 1
              ---------------
               C C C C C C C C
              ---------------
```

When all the switches are open (off), the display should look like this:

```
Switch:        8 7 6 5 4 3 2 1
              ---------------
               O O O O O O O O
              ---------------
```

> **NOTE**
> Return all switches to their original settings after testing.

3.5.15 **Character Attributes**

When this test is run, a status line displays reverse video, high-intensity, underlining, blinking, and double-wide characters. Color terminals display colored stripes; black, blue, green, cyan, red, magenta, yellow, and white from top to bottom. Red, green, blue (background), underline, blinking, double-wide, and red, breen and blue (foreground) attributes are displayed on the status line. On color terminal, reverse video does not apply.

3.5.16 **CRT Crosshatch Pattern**

When this test is run, a crosshatch pattern and the message "48.5 Hz Grid" will display on the screen. Pressing any key will display another crosshatch pattern and the message "61.5 Hz Grid." This test verifies the capability of the CRT monitor to hold vertical sync in 50 and 60 Hz vertical scan rate mode. Color terminals display only one grid and no message.

3.5.17 **CRT Brightness Pattern**

When this test is run, reverse video highlighted spaces are displayed, producing a standard "white level" screen. The color terminal displays a white screen. A red, green, blue, and black

screen and a white box on a black screen are displayed if "C" is consecutively pressed. Pressing any other key will return to the Diagnostic Menu.

3.5.18 Character Generator Test

This test displays all displayable characters, including block and bar graphics characters. Pressing any key will cause the following screens to be displayed: double-high characters, quad-size special characters and numbers, upper-case quad-size, lower-case quad-size, and process control graphics.

3.5.19 Touch Screen Test

When this test is run on workstations with touch screen, an 8 x 10 grid appears on the screen, dividing the screen into the 80 touch screen zones. When a touch screen zone is pressed, a block character should be displayed in that zone. When a zone is released, an R should be displayed in that zone.

The workstation displays the following message:

Touch each zone and 'R'elease each zone--Press any key to exit text

3.5.20 .CM Command

A hidden command, .CM, is available from the Diagnostic Menu for clearing the contents of the battery-backed CMOS RAM. This command clears the RAM, performs a resize of the RAM space, and initializes the real time clock to its default setting. To invoke this routine, simply type <.CM> at the Diagnostic Menu.

> **CAUTION**
> The .CM command will destroy any programs residing in CMOS RAM.

3.6 SET PASSWORD

A password may be used to tamper-proof the workstation's configuration.

To create, disable, or change a password, select #3, Set Password, from the Main Menu. The

following screen will be displayed:

Enter new password (3 characters)
or <ESC> to disable password
or <ENTER> to not change it:

To establish a password, enter 3 alphanumeric characters from the keyboard.

Once a password has been established, it is required to access the Main Menu after power-up. After striking F10 twice, the following prompt will be displayed:

Enter password (3 characters) or <ENTER> to quit:

Type the correct 3-character sequence and the Main Menu of the Set-up Mode will be displayed. When an incorrect password is entered the workstation remains in Operating Mode.

If the user forgets the password, the remote command Return Password will return the password to the host computer (see Chapter 6).

To change or disable the password, select item 3 from the Main Menu.

- To change the password, type any three alphanumeric characters and press the <Enter> key (the password will not be accepted until <Enter> is pressed). The password has now been changed. If a character was unintentionally pressed, you may use the backspace key (before <Enter> is pressed) to erase one or all three characters.

- To disable the password, press the <ESC> key without typing any other characters. (The password can be subsequently re-enabled by re-selecting item 3 from the Main Menu.)

- To retain the current password, press the <Enter> key without typing any characters.

The workstation configuration may also be tamper-proofed via keypad menu lockout. See Section 3.4.3.

3.7 TAB STOPS

> **NOTE**
> The tab stops are in effect only when the workstation is in ANSI mode, not in Hazeltine 1500 mode.

By selecting #4 from the Main Menu, the tab stops currently in effect will be displayed on the screen. Pressing the <TAB> key will send the cursor to the next tab stop.

The first row of numbers in the display are the column numbers. Below these column numbers (0 to 9 eight times, or a total of 80 columns) are the tab settings. An "S" below a number indicates a tab stop at that particular column position, while a blank beneath a number indicates no tab stop at that position. Tab settings at any column can be entered by using the cursor keys to move the cursor under the desired column, then typing S. Tab settings can be removed by moving the cursor to the desired column and typing a space. Up to 80 tabs (one for each column) can be entered.

If ANSI remote commands are used to change the tab stops, these changes will not be saved if the terminal is powered-down or reset. The terminal will be reinitialized to the settings in the Configuration Menu.

3.8 STORED SCREEN UTILITIES MENU

This option from the Main Menu is used to create new program blocks and manipulate existing ones. Figure 3-10 displays the following menu which will appear:

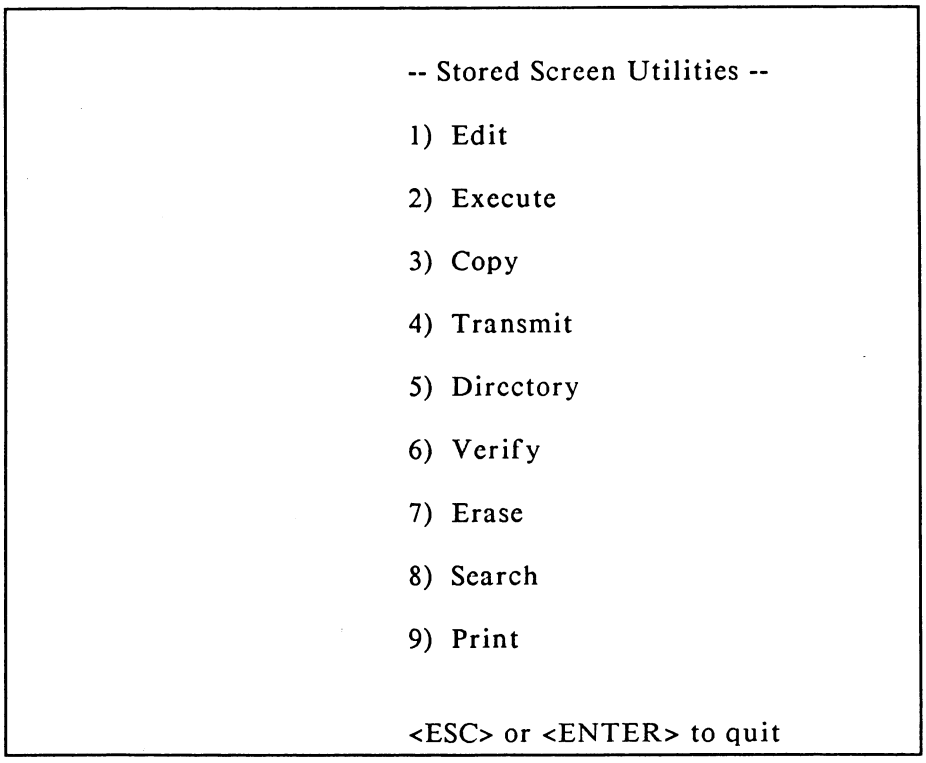


Figure 3-10. Stored Screen Utilities Menu

Menu options are summarized below and explained in more detail in the following setions.

Edit. Enter a new screen or edit an existing screen. The user is prompted for the number of the screen to edit (1-255). See section 3.8.1.

Execute. Prompts the user for the number of the screen to be executed. See section 3.8.2.

CAUTION
When copying one screen to another, any data in the destination screen will be lost, replaced entirely by the newly copied data.

Copy. Copies one stored screen to another. The user is prompted for the source screen (to be copied) and the destination screen (into which the source screen will be copied). See section 3.8.3.

Transmit. Transmits a stored screen to another terminal or system, via the printer port of the 2000. The user is prompted for the number of the screen to transmit (or all screens), and the number of the destination screen. See section 3.8.4.

Directory. Displays the first line of each stored screen. See section 3.8.5.

Verify. Compares a downloaded image with the contents of a specified stored screen or all stored screens, via the standard serial port. This option is useful if option #4 (Transmit Screen) was used to backup screen programs to another computer system. See section 3.8.6.

Erase. Erases any screen or range of screens of the 255 screens in memory. See section 3.8.7.

Search. Searches for a specified string of characters, up to 20 characters in length. All screens, starting at screen #1, are searched. If a match is found, the screen displays the string, the screen number, and the line number in which the string was found. The user may either quit or continue searching. See section 3.8.8.

Print. Prints a specified screen or range of screens, via the printer port. See section 3.8.9.

Notice there are no commands to create or save stored screens. A new terminal has 255 "empty" stored screen programs. When a stored screen program is edited, data is inserted or deleted from the screen, but the screen program itself will always stay intact even if a power failure occurs during editing.

When using the Stored Screen Utilities, the status area may display the following:

- Prompts for user input
- List of commands
- Unused bytes remaining in the CMOS RAM allotted for screen (in decimal)
- Error messages

3.8.1 Using the Editor

> **NOTE**
> Entering and editing screens requires the use of an external full-stroke or membrane AT- or XT-style keyboard. See Figure 2-9 for dipswitch settings.

When edit is selected from the Stored Screen Utilities Menu, the workstation is put in edit mode. In this mode new screens are typed and existing screens are modified. Before describing the operation of the editor, it is necessary to define a few terms:

Screen Block — A screen block is defined as the memory space occupied by a single screen. 255 screen blocks are available, and each of them can store a separate screen. On a new workstation with a new 2000-01 firmware chip, all of the screen blocks are empty.

Screen — A screen consists of a sequence of instructions. Each command gives the workstation specific instructions concerning exactly what characters or symbols to display on the screen, and where on the screen these characters will be displayed. You can also select the size of the characters (single, double, or quad size) and their attributes (blinking, reverse video, etc.).

Screen Execution — Merely entering a screen into a screen block does not create a screen display. The screen program will remain in battery-backed CMOS memory for an indefinite period of time (or until it is replaced by a different program typed or copied into the same block or erased). However, the screen program is created only when the program is executed. It will remain on the screen until another screen is executed or until you enter Set-Up Mode to reconfigure the workstation.

Upon entering the Editor, the following prompt appears at the bottom of the screen:

Edit which screen? (1-255 or <ESC>):

Enter the number of the screen you wish to create or edit, press <ENTER>, and the program block appears on the screen. The status line at the bottom of the edit screen looks like this:

<DEL>Char <ALT><DEL>Line <F5>Quit <F6>Execute Scr # 1 Free:28615

To enter a screen into a screen block, simply type the program. The <ENTER> key will move the cursor to the beginning of the next line. The built-in screen editor allows for correction by deleting and inserting characters.

The text of a screen can only be altered at the cursor position. Keys which control cursor movement are defined in Table 3-2, below.

Table 3-2. Cursor Movement Keys

Cursor Movement Keys	
→	Moves cursor one position to the right
←	Moves cursor one position to the left
↓	Moves cursor one line down
↑	Moves cursor one line up
<END>	Moves cursor to the end of the line
<HOME>	Moves cursor to the beginning of the line
<PG UP>	Moves cursor to the top of the page
<PG DN>	Moves cursor to the bottom of the page

Control characters are displayed with a two-character representation of these ASCII names. Thus the carriage return at the end of each line is displayed as a "CR" symbol.

Inserting Characters. The editor is always in "insert" mode. This means that any characters typed will be inserted into the program block at the cursor position. Because of this, a user can never inadvertently overwrite anything. A specific action must be made to replace or delete any text.

Deleting Characters. Characters are deleted at the cursor position with the <DEL> key; characters are deleted to the left of the cursor with <BACKSPACE>.

Replacing Characters. To replace a character, delete the character to be replaced and insert the new character. For example, to replace the "O" in

WORNING_

with an "A", first move the cursor to the "O",

WORNING

then press the <DEL> key once to delete the "O",

WRNING

and finally type (uppercase) "A" to insert the "A"

WARNING

Inserting a Line. To insert a new line or break a single line into two lines, move the cursor to that point and press the <RETURN> key. The character at the cursor position will now be the first character of the next line, and the rest of the lines in the program block (if any) will be moved down one line. A carriage return symbol will appear at the end of each line.

Deleting a Line. To delete an entire line, press the <ALT> and <DEL> keys simultaneously. This deletes the line where the cursor is currently positioned.

Testing a Screen. A newly-entered screen can be tested immediately. To execute the screen block displayed on the screen, press <F6>. (Before executing the program, the workstation automatically clears the screen, puts the cursor in the upper left hand corner, and resets all display attributes to their default condition.)

The screen will display an error message if any commands have been typed incorrectly. The line on which the first error occurred will be near the top of the screen, with the cursor positioned where the program was executing when the error was detected. (The error may not be precisely at the cursor location, but it will be close.) To display the entire program block again, move the cursor upward to scroll the screen.

If <F10> is pressed twice during execution, the workstation will stop the program and return to the editor.

3.8.1.1 Sample Stored Screen

A typical stored screen program is shown below. When executed, this screen displays the letters "WARNING" in a box.

Both the Hazeltine 1500 and ANSI versions of the screen program are included. To execute a stored screen written in Hazeltine 1500 format, the workstation must be configured for Hazeltine 1500 emulation via the Miscellaneous Configuration Menu (see Section 3.4.3). Likewise, to execute a screen written in ANSI format, the workstation must be configured for ANSI emulation.

Screen Program Example

Hazeltine 1500

```
   7E 09 01 12 05 36 0A      ;Draw Box command<CR>
   7E 11 13 06               ;Cursor to X,Y command to put cursor in box<CR>
   7E 03 09 02               ;Change Char. Attributes command to quad,<CR>
                             ;blink, reverse<CR>
   "WARNING"                 ;message in box<CR>
```

ANSI

```
<ESC>"[2;1;6;19;11;55p"      ;Draw Box command<CR>
<ESC>"[7;20H"                ;Cursor to X,Y command to put cursor in box<CR>
<ESC>"[1;9;2p"               ;Change Char. Attributes command<CR>
"WARNING"                    ;message in box<CR>
```

If displayable characters or remote commands are encoded in ASCII, the sequence of ASCII characters must be enclosed in quotation marks ("). This is because all characters are interpreted as hex, unless enclosed in quotation marks. It is recommended that each separate line be enclosed in quotes, as in the above example. If the above example were enclosed in only two quotes, one at the beginning and one at the end, the <CR>s at the end of each line would be interpreted as remote commands and executed and the comments would be printed, thereby affecting the current cursor position. If each line is embedded in quotes, the <CR> at the end of each line is not interpreted as a remote command.

Note also that it is not required that Hazeltine-format screens be entered in hex and ANSI-format screens in ASCII. Hazeltine-format commands may also be entered in ASCII (if the commands are enclosed in quote marks), and ANSI-format commands in hex. However, since ANSI emulation is designed so that ASCII characters can be used, ANSI-format commands will typically be entered in ASCII.

Likewise, Hazeltine-format commands will typically be entered in hex. Note that commands entered as ASCII characters generally take up less program storage than hex codes, since it takes two hex digits to reproduce a single ASCII character.

The <CR> and <ESC> shown above actually take up only one character location each on the screen. <CR> is visible so the user can see the end of the line.

Characters representing hex bytes (i.e., 7E 01 07) must always appear in pairs. For example, 1 is not a legal representation of the hex byte 01. The leading zero is required.

Spaces not enclosed in quotes are ignored. For example, either 7E 01 3E or 7E013E is an acceptable way of representing 3 hex bytes.

Any text appearing between quotes (") is stored as ASCII characters, and is used literally. All other text represents hexadecimal bytes. A double quote ("") that appears between quotes will cause a single quote to be displayed when the screen is executed.

Any text following a semicolon (;) outside of quotes is a comment. Comments extend to the end of the line. When a stored screen is executed, comments are ignored.

It is a good idea to include an identifying comment as the first line of each stored screen so that it will show up in the directory entry for the screen.

3.8.1.2 **Nested Screens**

The remote command Execute Stored Screen can be included **within** a screen program. For example, suppose screen mm contains an Execute Stored Screen nn command. When Screen mm is executed, all the remote commands will be executed in the order in which they occur. When the Execute Stored Screen nn command is executed, control will be transferred to Screen nn, which will be executed in its entirety. When Screen nn is finished, control will be returned to Screen mm, and execution of subsequent commands in screen mm will continue.

Any number of Execute Stored Screen commands can be included in one screen, and they may be included anywhere within the screen. Up to ten levels of nesting is permitted.

Stored Screens are especially useful in designing complicated displays in which some portions of the display vary while others remain the same. The constant areas of a display can be put in one screen, while each of the possible variations may be placed in separate screens. These variations can be called from other screens as needed. Then instead of redesigning each screen from scratch, the user may design a screen in parts and reuse these parts in other screens.

A simple example is shown below. Suppose an application is monitoring a vat for two danger conditions, "pressure too low" and "temperature too high." When it detects either condition, the application should flash "WARNING" at the top of the screen. However, it would also be desirable to identify the type of danger condition by displaying the letters "Temperature Too High" or "Pressure Too Low" following the "WARNING" message. One way of designing the screens for this application is to use three separate screens (see Figure 3-11).

Screen 24: Creates the display "WARNING"
Screen 20: Creates the display "TEMPERATURE TOO HIGH"
Screen 21: Creates the display "PRESSURE TOO LOW"

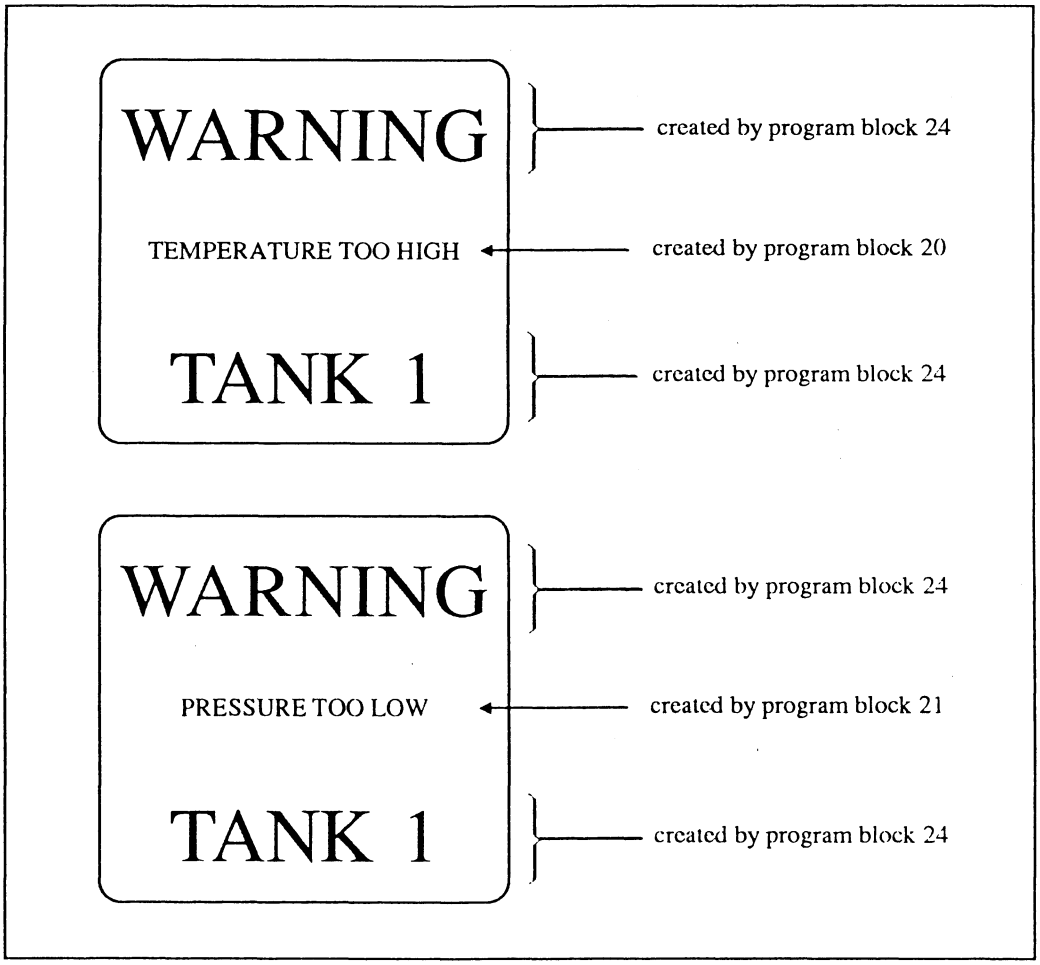


Figure 3-11. Nested Screen Example

To display the "TEMPERATURE TOO HIGH" warning, screen 7 would be executed.

To display the "PRESSURE TOO LOW" warning, screen 8 would be executed. Screen 7 contains an Execute Stored Screen command to execute Screens 24 and 20, while screen 8 executes Screens 24 and 21.

Sample Nested Screen

```
Screen 7 (For "TEMPERATURE TOO HIGH" Warning on Tank 1)
   7E 10 18          ;Execute Stored Screen 18H (=24)
   7E 10 14          ;Execute Stored Screen 14H (=20)

Screen 8 (For "PRESSURE TOO LOW" Warning on Tank 1)
   7E 10 18          ;Execute Stored Screen 18H (=24)
   7E 10 15          ;Execute Stored Screen 15H (=21)

Screen 24
   7E 11 1D 03        ;Cursor to X,Y command
   7E 03 1A 01        ;Blinking, double-size, high-intensity
   "WARNING"
   7E 11 1E 11        ;Cursor to X,Y command
   7E 03 02 01        ;Double-size, high-intensity
   "TANK 1"
   7E 11 1B 0B        ;Cursor to X,Y command
   7E 03 00 00        ;Default Character Attributes

Screen 20
   "TEMPERATURE TOO HIGH"

Screen 21
   "PRESSURE TOO LOW"
```

Using the Primary Serial Port for Input During Screen Program Execution

During execution of a screen program, the workstation monitors the RS-485 Primary Serial Port, storing any characters received in a serial data buffer. When the last character in the screen program has been read, the workstation proceeds to read and execute the serial data buffer. The sample screen on the following page uses this feature to allow the host to select the next screen to be executed:

Sample Screen

```
Screen 1
    7E 1C                          ;Clear Screen
    7E 11 1D 03                    ;Cursor to X,Y
    7E 03 1A 01                    ;Change Character Attributes
    "WARNING"
    7E 11 1E 11                    ;Cursor to X,Y
    7E 03 12 01                    ;Change Character Attributes
    "TANK 1"
    7E 11 1B 09                    ;Cursor to X,Y
    7E 03 00 00                    ;Change Character Attributes
    7E 10                          ;First part of Execute Screen command,
                                   ;waiting for last byte from host system

Screen 2
    "TEMPERATURE TOO HIGH"
Screen 3
    "PRESSURE TOO LOW"
```

Whatever character is received by the workstation over the serial line will determine the next screen to be executed. For example, if 02H is received, Screen 2 will be executed.

Entering Variable Data into a Stored Screen

Nested screens can be used to display changing data in a screen display. For example, suppose that the message to be displayed on the screen is "Temperature of Tank 2 is xxx Degrees", where xxx varies. The following screens could display the message:

```
Screen 10
    7E 1C                          ;Clear Screen
    "Temperature of Tank 2 is "
    7E 10 0B                       ;Execute Screen 11
    " Degrees"

Screen 11
    Contains a number written by the host
```

The host would periodically write new data to screen 11 by issuing a Receive Screen remote command. For example, to transmit the number "120" to screen 11, the host would transmit the following hexadecimal data to the terminal:

```
7E 1E 0B 22 31 32 30 22 7F
```

Note that 22 is the ASCII value of the quotation mark (").

3.8.2 **Screen Execution**

A screen can be executed in four ways:

- from within the editor, by pressing the <F6> key;
- by selecting option #2 - "Execute" from the Stored Screen Utilities Menu;
- by selecting the start-up program in the Misc Menu;
- by issuing a remote command (discussed further in Chapter 6).

If option #2 is selected from the Stored Screen Utilities Menu, a prompt asks for the number of the screen to be executed:

Execute which screen? (1-255 or <ESC>):

Consult the directory (option #5 of the Stored Screen Utilities Menu) if the screen number is not known (for more information on the directory, see Section 3.8.5). Press <Esc> to exit the "Execute" menu without executing a screen.

If a screen is executed from the editor and an error is encountered, an error message is displayed. If the screen is executed from a remote command, and an error occurs, screen execution stops, but no error message is displayed.

For all methods, before the workstation executes a screen, it automatically clears the screen, puts the cursor in the upper left hand corner, and resets all display attributes to their default condition.

To exit from the executing screen, press <F10> twice.

3.8.3 Copying Screens

Option #3, "Copy" from the Stored Screen Utility Menu allows the user to copy any screen to any of the 255 screen blocks in memory. This utility provides a means for arranging screens in a specific numerical order or for assigning certain types of screens to certain blocks of numbers for indexing purposes. When option #3 is selected from the Stored Screen Utilities Menu the following prompts will be displayed:

Copy from which screen? (1-255 or <ESC>):

and then

Copy into which screen? (1-255 or <ESC>):

and then

Number of screens to copy? (1-255 or <ESC>):

Example:

Copy from which screen? **10**
Copy into which screen? **50**
Number of screens to copy? **2**

Screen 10 will be copied into screen 50, and screen 11 will be copied into screen 51.

> **CAUTION**
> All data in the destination screen is lost during a copy.

To prevent the loss of existing screens, consult the directory prior to making a copy.

3.8.4 Transmit Screens

Option #4 from the Stored Screen Utilities Menu, "Transmit Screens," is provided as a means of saving and loading screen programs via the serial printer port to another terminal or PC with a software program such as PROCOMM. This command sends a Receive Stored Screen command, followed by the actual screen, followed by a DEL (7FH) out the printer port for each screen specified.

When this option is selected the following prompt is displayed:

Transmit all screens? (Y,N or <ESC>):

If yes is selected, the transmit begins. All 255 screens, including empty screens will be transmitted.

If no is selected, the following prompt is displayed:

Transmit (1-255 or <ESC>):

After selecting the screen number, hit <Enter>. The following prompt will appear:

Transmit it as which screen? (1-255 or <ESC>):

After selecting the destination screen number, hit <Enter> to receive the following prompt:

Number of screens to Transmit (1-255 or <ESC>):

After selecting the number of screens (in sequence) to transmit, hit <Enter>. When the transmission is complete the following message will appear:

Screen transmission done. Press any key to continue.

3.8.4.1 Backing Up Screens on the IBM PC

Screens tored on a Xycom 2000 can be backed up to an IBM PC with any communication package that supports ASCII transfer. To back up screens, follow the steps below:

1. Configure the software package for baud rate, data bits, and handshaking to match the 2000 Printer Port Configuratiopn menu.

2. Select ANSI or Hazeltine emulation from the Miscellaneous Configuration Menu on the 2000.

3. When the software package is ready to receive an ASCII file, select Transmit on the 2000 and follow the instructions on the screen.

The screens will be sent to the IBM PC with the correct ANSI or Hazeltine remote commands embedded in the file.

To restore saved screens, put the 2000 in terminal mode and send the previously saved screens from the IBM PC to the 2000.

To connect the 2000 to an IBM PC make the connections shown below:

9-Pin IBM PC	**2000 (9-pin)**	**or**	**25-Pin IBM PC**
3 (TXD)	2 (RXD)		2 (TXD)
2 (RXD)	3 (TXD)		3 (RXD)
5 (GND)	5 (GND)		7 (GND)

3.8.5 Directory

Option #5 "Directory" from the Stored Screen Utilities Menu allows the user to see the first line of every program block in screen memory. Thus, it is recommended that the first line of every program be a program name or title (any text proceeded by a semi-colon will be treated as a comment or remark and will be ignored). The directory option displays the program blocks in groups of 20. Exit the directory by pressing <Esc> or <Enter>. The next group of 20 blocks can be accessed by pressing any other key.

3.8.6 Verify

Option #6 "Verify" from the Stored Screen Utilities Menu allows the user to check data just transferred against the original program in memory (in order to verify that the transfer was successful).

When this option is selected, the following prompt will appear:

Verify all screens? (Y,N or <ESC>):

If yes is selected, verification begins. If no is selected, the following prompt will appear:

Verify which screen? (1-255 or <ESC>):

After the screen to verify is selected, verification begins. If an error occurs during verification, the workstation will immediately display a "Verification failed" message. After a successful verification, a "Verification passed" message will appear.

3.8.7 **Erase**

Option #7 "Erase" from the Stored Screen Utilities Menu allows the user to erase any screen or range of screens of the 255 screens in memory. When this option is selected from the Program Utilities Menu, the following prompt will be displayed:

Erase starting at which screen? (1-255 or <ESC>):

and then

Number of screens to erase? (1-255 or <ESC>):

and then

OK to erase screens x to y ? (Y or N):

Example:
Erase starting at which screen? **10**
Number of screens to erase? **2**
OK to erase screens 10 to 11 ? (Y or N): **Y**

Screens 10 and 11 will be erased.

3.8.8 **Search**

Option #8 "Search" from the Stored Screen Utilities Menu allows the user to search for a string of characters. The user may specify a string up to twenty characters in length. All program blocks, starting at block #1, are searched. If a match is found, the screen displays the string, the program block number, and the line number in which the string was found. Press <ESC> to quit or C to continue searching.

3.8.9 **Print**

Option #9 "Print" from the Stored Screen Utilities Menu takes the user through the same sequence of prompts for screen numbers as the transmit function. When screen selection is complete, the screens will be output to the selected serial printer port. Only non-empty screens will be printed.

> **NOTE**
> When the parallel printer is selected in the Miscellaneous Configuration Menu, the Centronics parallel port is used for output to the printer.

4.1 KEYBOARDS

Several different keyboards may be used with the 2000 Series Industrial Workstations, including a full-stroke PC/AT or XT style keyboard, a full sealed QWERTY style keyboard with 20 function keys, and a sealed 58-key ABC style keypad with 20 function keys. Generally, only a sealed membrane keyboard should be used when the workstation is serving as a plant-floor operator interface. A full-stroke full-size keyboard is usually used for programming and supervisory operations.

The keyboards generate hexadecimal and ASCII codes when a key or combination of keys is pressed. Codes returned differ depending on whether Hazeltine or ANSI mode is used.

4.1.2 Keyboard Codes

All three keyboards return identical codes.

> **NOTE**
> When the Menu Entry Lockout is enabled, the sealed keyboard F10 key will **not** respond.

The hexadecimal and ASCII codes generated by the keyboards are listed in Tables 4-1 and 4-1A.

Table 4-1. Keyboard Key Codes (Full and Half-duplex)[1]

	no CTRL, no SHIFT		no CTRL, SHIFT		CTRL, no SHIFT		CTRL, SHIFT	
Key	Hex	ASCII	Hex	ASCII	Hex	ASCII	Hex	ASCII
A	61	a	41	A	01	<SOH>	01	<SOH>
B	62	b	42	B	02	<STX>	02	<STX>
C	63	c	43	C	03	<ETX>	03	<ETX>
D	64	d	44	D	04	<EOT>	04	<EOT>
E	65	e	45	E	05	<ENQ>	05	<ENQ>
F	66	f	46	F	06	<ACK>	06	<ACK>
G	67	g	47	G	07	<BEL>	07	<BEL>
H	68	h	48	H	08	<BS>	08	<BS>
I	69	i	49	I	09	<HT>	09	<HT>
J	6A	j	4A	J	0A	<LF>	0A	<LF>
K	6B	k	4B	K	0B	<VT>	0B	<VT>
L	6C	l	4C	L	0C	<FF>	0C	<FF>
M	6D	m	4D	M	0D	<CR>	0D	<CR>
N	6E	n	4E	N	0E	<SO>	0E	<SO>
O	6F	o	4F	O	0F	<SI>	0F	<SI>
P	70	p	50	P	10	<DLE>	10	<DLE>
Q	71	q	51	Q	11	<DC1>	11	<DC1>
R	72	r	52	R	12	<DC2>	12	<DC2>
S	73	s	53	S	13	<DC3>	13	<DC3>
T	74	t	54	T	14	<DC4>	14	<DC4>
U	75	u	55	U	15	<NAK>	15	<NAK>
V	76	v	56	V	16	<SYN>	16	<SYN>
W	77	w	57	W	17	<ETB>	17	<ETB>
X	78	x	58	X	18	<CAN>	18	<CAN>
Y	79	y	59	Y	19	<EM>	19	<EM>
Z	7A	z	5A	Z	1A	<SUB>	1A	<SUB>
1	31	1	21	!	11	<DC1>	01	<SOH>
2	32	2	40	@	12	<DC2>	00	<NULL>
3	33	3	23	#	13	<DC3>	03	<ETX>
4	34	4	24	$	14	<DC4>	04	<EOT>
5	35	5	25	%	15	<NAK>	05	<ENQ>
6	36	6	5E	^	16	<SYN>	1E	<RS>
7	37	7	26	&	17	<ETB>	06	<ACK>
8	38	8	2A	*	18	<CAN>	0A	<LF>
9	39	9	28	(	19	<EM>	08	<BS>
0	30	0	29	)	10	<DLE>	09	<HT>
<BACKSP>	08	<BS>	08	<BS>	08	<BS>	08	<BS>
<ESC>	1B	<ESC>	1B	<ESC>	1B	<ESC>	1B	<ESC>
<SPACE>	20	Space	20	Space	00	<NUL>	00	<NUL>
'	27	'	22	"	07	<BEL>	02	<STX>

Table 4-1. Keyboard Key Codes (Full and Half-duplex) Continued

	no CTRL, no SHIFT		no CTRL, SHIFT		CTRL, no SHIFT		CTRL, SHIFT	
Key	**Hex**	**ASCII**	**Hex**	**ASCII**	**Hex**	**ASCII**	**Hex**	**ASCII**
*	2A	*						
<PRT SCN>	AA	*	AA	N/A	0A	<LF>	0A	<LF>
,	2C	,	3C	<	0C	<FF>	1C	<FS>
-	2D	-	5F	_	0D	<CR>	1F	<US>
.	2E	.	3E	>	0E	<SO>	1E	<RS>
/	2F	/	3F	?	0F	<SI>	1F	<US>
;	3B	;	3A	:	1B	<ESC>	1A	<SUB>
=	3D	=	2B	+	1D	<GS>	0B	<VT>
[	5B	[	7B	{	1B	<ESC>	1B	<ESC>
\	5C	\	7C	|	1C	<FS>	1C	<FS>
]	5D	]	7D	}	1D	<GS>	1D	<GS>
`	60	`	7E	~	00	<NUL>	1E	<RS>
<DEL>	7F	<DEL>	2E	.	1F	<US>	0E	<SO>
<PAUSE>	BC							
<CLEAR>	BD							
<HOME>	B7							
<END>	B1							
<PGUP>	B9							
<PGDN>	B3							
<BKSPC>	08							
<INSERT>	B0							
<ENTER>	0D	<CR>						
<TAB>	09	<HT>						
F1	80							
F2	81							
F3	82							
F4	83							
F5	84							
F6	85							
F7	86							
F8	87							
F9	88							
F10	8A							
dF11	8B							
F12	8C							
F13	8D							
F14	8E							
F15	8F							
F16	90							
F17	91							

Table 4-1. Keyboard Key Codes (Full and Half-duplex) - Continued

	no CTRL, no SHIFT		no CTRL, SHIFT		CTRL, no SHIFT		CTRL, SHIFT	
Key	**Hex**	**ASCII**	**Hex**	**ASCII**	**Hex**	**ASCII**	**Hex**	**ASCII**
F18	92							
F19	93							
F20	94							

Table 4-1A. Keyboard Key Codes (Application Mode)

Key	Hzltn Code	ANSI Code
F1	80	<ESC>OP
F2	81	<ESC>OQ
F3	82	<ESC>OR
F4	83	<ESC>OS
F5	84	<ESC>OT
F6	85	<ESC>OU
F7	86	<ESC>OV
F8	87	<ESC>OW
F9	88	<ESC>OX
F10	8A	<ESC>OY
F11	8B	<ESC>OZ
F12	8C	<ESC>O[
F13	8D	<ESC>O\
F14	8E	<ESC>O]
F15	8F	<ESC>O^
F16	90	<ESC>O _
F17	91	<ESC>O'
F18	92	<ESC>Oa
F19	93	<ESC>Ob
F20	94	<ESC>Oc
Up arrow	7E OC	<ESC>[OA
Down arrow	OA	<ESC>[OB
Left arrow	O8	<ESC>[OD
Right arrow	10	<ESC>[OC
PgUp	B9	<ESC>O B9(Hex)
PgDn	B3	<ESC>O B3(Hex)
Period	AE	<ESC>On
Minus	AD	<ESC>Om

Table 4-1A. Keyboard Key Codes (Application Mode) - Continued

Key	Hzltn Code	ANSI Code
ENTER	8D	<ESC>OM
BkSp	O8	<ESC>O O8 (hex)
Del	O7	<ESC> O 7F(Hex)
Home	O	<ESC> O B7(Hex)
ESC	9B	<ESC> O <ESC>
End	B1	<ESC>OB1(Hex)
0	BO	<ESC>Op
1	B1	<ESC>Oq
2	B2	<ESC>Or
3	B3	<ESC>Os
4	B4	<ESC>Ot
5	B5	<ESC>Ou
6	B6	<ESC>Ov
7	B7	<ESC>Ow
8	B8	<ESC>Ox
9	B9	<ESC>Oy
A	C1	<ESC>Oa
B	C2	<ESC>Ob
C	C3	<ESC>Oc
D	C4	<ESC>Od
E	C5	<ESC>Oe
F	C6	<ESC>Of
PF1	95	<ESC>O 95(Hex)
PF2	96	<ESC>O 96(Hex)
PF3	97	<ESC>O 97(Hex)
PF4	98	<ESC>O 98(Hex)
PF5	99	<ESC>O 99(Hex)
PF6	9A	<ESC>O 9A(Hex)
PF7	9B	<ESC>O 9B(Hex)
PF8	9C	<ESC>O 9C(Hex)

Several keyboard keys return different codes when firmware containing Operator Interface Language (OIL) is installed. These keys and codes are listed in Table 4-2, on the following page.

Table 4-2. Keyboard Key Codes Unique to OIL

Key	Hex	ASCII
Up Arrow	11	<DC1>
Left Arrow	12	<DC2>
Right Arrow	13	<DC3>
Down Arrow	14	<DC4>

4.2 TOUCH SCREEN

A touch screen option, 2000T, is available for the 2000 Industrial Workstation. This option allows the user to transmit codes or screens from the touch screen.

The touch screen is a clear panel, superimposed on the Lexan shield of the 2000 Industrial Workstation, that divides the screen into 80 zones. It is virtually transparent to the user.

4.2.1 Default Touch Screen Codes

2000 Industrial Workstations with touch screen can operate in two modes: Normal Mode and Touch Screen Mode. In **Normal Mode**, a touch screen zone will transmit a hexadecimal code when pressed. Touch screen zones and the default codes they return are listed in Figure 4-1 on the following page. When a touch screen zone is released, it will transmit a 0. In **Touch Screen Mode**, these codes are preceded by ESC T.

NOTE
When OIL firmware is installed on the 2000 Industrial Workstation, touch screen zones return codes other than those listed in Figure 4-1. See the 2000 OIL Manual for more information.

NOTE
On power-up the 2000 with touch screen is always in Normal Mode.

Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9	Z10
80H	81H	82H	83H	84H	85H	86H	87H	88H	89H
Z11	Z12	Z13	Z14	Z15	Z16	Z17	Z18	Z19	Z20
8AH	8BH	8CH	8DH	8EH	8FH	90H	91H	92H	93H
Z21	Z22	Z23	Z24	Z25	Z26	Z27	Z28	Z29	Z30
94H	95H	96H	97H	98H	99H	9AH	9BH	9CH	9DH
Z31	Z32	Z33	Z34	Z35	Z36	Z37	Z38	Z39	Z40
9EH	9FH	A0H	A1H	A2H	A3H	A4H	A5H	A6H	A7H
Z41	Z42	Z43	Z44	Z45	Z46	Z47	Z48	Z49	Z50
A8H	A9H	AAH	ABH	ACH	ADH	AEH	AFH	B0H	B1H
Z51	Z52	Z53	Z54	Z55	Z56	Z57	Z58	Z59	Z60
B2H	B3H	B4H	B5H	B6H	B7H	B8H	B9H	BAH	BBH
Z61	Z62	Z63	Z64	Z65	Z66	Z67	Z68	Z69	Z70
BCH	BDH	BEH	BFH	C0H	C1H	C2H	C3H	C4H	C5H
Z71	Z72	Z73	Z74	Z75	Z76	Z77	Z78	Z79	Z80
C6H	C7H	C8H	C9H	CAH	CBH	CCH	CDH	CEH	CFH

Figure 4-1. Default Codes Transmitted by Touch Screen Zones (Base Terminal)

KEY:

Z = Zone (e.g., Z1 = touch screen zone 1)

4.2.2 Touch Screen Remote Commands

> **NOTE**
> When OIL firmware is installed on the 2000 Industrial Workstation, the following remote commands are not valid. Touch Screen OIL commands are listed in Chapter 4 of the 2000 OIL manual.

4.2.2.1 Changing Modes

To toggle between Normal Mode and Touch Screen Mode, use the following remote commands (these commands are also listed in Chapter 6 - Remote Commands):

Command: Put 2000 Workstation in Normal Mode

ANSI: ESC [41; 0 p

Hazeltine: 7E 3A O0 (Hex)
~ : <NUL> (ASCII)

Command: Put 2000 Workstation in Touch Screen Mode
(transmitted codes will be preceded by "ESC T")

ANSI: ESC [41; 1 p

Hazeltine 7E 3A O1 (Hex)
~ : <SOH> (ASCII)

4.2.2.2 Set Programmable Touch Screen Zone

A touch screen zone may be configured to transmit a stored screen. To do this use the following remote command (this command is also listed in Chapter 6 - Remote Commands):

Command: Set programmable touch screen zone

ANSI: ESC [40;<zone>;<screen> p

Hazeltine: 7E 39 <zone> <screen> (Hex)
~ 9 <zone> <screen> (ASCII)

where: <zone> is the touch screen zone to be configured
<screen> is the screen to send (1 - 255)

4.2.2.3 **Define Zone or Zones**

A zone or contiguous area of zones may be set to return a common user-specified code other than the default code. To do this, use the following remote command (this command is also listed in Chapter 6 - Remote Commands).

Command: Define zone or zones

ANSI: ESC [42; <upper left zone>;<lower right zone>;<code> p

Hazeltine 7E 3B <upper left zone> <lower right zone> <code> (Hex)
~ ; <upper left zone> <lower right zone> <code> (ASCII)

where: <upper left zone> is the upper left boundary of the zone to be defined
<lower right zone> is the lower right boundary of the zone to be defined
<code> is a decimal (1-255) or hexadecimal (1-FF) number

Example 1: ESC [42; 35;47;12 p

This example causes contiguous touch screen zones 35-37 and 45-47 to return the value 12 when pressed. Other zones remain unaffected.

Example 2: ESC [42; 71;71;BB p

This example causes the single touch screen zone 71 to return the hexadecimal value BB when pressed. Other zones remain unaffected.

4.2.3 **Hierarchy of Touch Screen Zone Returns**

Regardless of the mode the workstation is operating in, a touch screen zone configured to return a stored screen via the "set programmable touch screen zone" remote command will return the assigned screen rather than a code. If no screen is assigned to a given zone, the zone will return a code. If the touch screen zone has been assigned a code with the remote command "define zone or zones", that code will be returned. If no code has been assigned to the touch screen zone, the default code (listed in Table 4.1) will be transmitted. When operating in Touch Screen Mode, codes returned are preceded by "ESC T."

5.1 VIDEO DISPLAY FORMAT

Table 5-1. 2000 Series Industrial Workstations Video Display Format

CHARACTERTISIC	SPECIFICATIONS		
Features	**2000/2005**	**2060**	**2050**
Screen Size	9" diagonal	12" diagonal	
Screen Phosphor	Amber		Color
Screen Capacity	25 rows x 80 columns (standard characters) 25 rows x 40 columns (double-wide characters) 12 rows x 80 columns (double-high characters) 12 rows x 40 columns (double-size characters) 6 rows x 13 columns (quad-size characters) (2050) 6 rows x 16 columns (quad-size characters)		
Cell Size	10 pixels wide by 12 pixels high		8 pixels wide by 10 pixels high
Character Size	1 cell (regular characters) 5 cells wide by 4 cells high (quad-size characters) 6 cells wide by 5 cells high (quad-size characters) (2050) 4 cells wide by 4 cells high (process-control characters)		
Character Set	See appendix B for the graphic characters		
Character Attributes	blink underline double-wide double-high double-size quad-size reverse video high intensity		blink underline double-wide quad-size double-size 8 fore-ground & background colors
Remote Commands	variety of commands to draw boxes, vertical or horizontal lines, and high-resolution bars		
Cursor	blinking underline, blinking block, or none		

5.2 CURSOR ADDRESSING

Each workstation provides two cursor-addressing commands: Cursor To and Return Cursor Position. One of these -- Cursor To X,Y -- allows the cursor to be positioned anywhere on the

video display. The other -- Return Cursor Position -- allows the current position of the cursor to be read.

The video display has a coordinate system for cursor positioning. Diagrams showing row and column coordinates for each possible cursor position are given in Figures 6-1 and 6-2 in the next chapter. Row and column coordinates begin with 0 in Hazeltine mode, and 1 in ANSI mode. When the cursor address is read, the system will return its column and row coordinate. Those coordinates are also used to move the cursor.

> **NOTE**
> In Hazeltine emulation, the column coordinates precedes the row coordinate (x, y); in ANSI emulation the row coordinate precedes the column (y, x). Also, column and row coordinates are different for Hazeltine and ANSI emulation.

The character sequence required to execute the Cursor To X,Y and Return Cursor Position commands depends upon whether the workstation is configured for Hazeltine or ANSI emulation. Chapter 6 contains these commands.

5.3 STATUS AREA

Up to 24 status lines may be selected from the Miscellaneous Configuration Menu (discussed in Section 3.4.3). The status area is not affected by the action of the normal display area (i.e., it does not scroll and the status line is not cleared when the screen is cleared). In order to write to the status line, a Cursor to X,Y remote command must be sent to the workstation to move the cursor to the desired row of the status area. Remote commands positioned in the status area will not affect the normal display area. Therefore, a clear screen command executed while in the status area will only clear the status area.

5.4 SCROLLING

The video display will scroll up whenever any of the following conditions exist:

- the cursor is in the last character position of the bottom line (line 24), when scrolling is enabled, autowrap is enabled, and a displayable ASCII code is received
- the cursor is in the bottom line, auto linefeed is enabled, and a carriage return <CR> code is received
- the cursor is in the bottom line, auto linefeed is disabled, and a linefeed code is received
- any "move cursor with scrolling" commands are received

When the display scrolls up, the top line of the display is removed, all lines on the display except the status line shift up one line, a blank line is added immediately above the status line, and the cursor is moved to the new line. This new line consists of background spaces.
If scrolling is turned off, (either through the Configuration Menu or a remote command), the video display will not scroll.

5.5 ATTRIBUTES

The appearance of characters on the display can be enhanced by assigning attributes to characters. In addition, the attributes can be used to select alternate character sets or sizes of characters. Attributes that can be assigned to characters are:

- blinking characters
- underline
- regular, double-wide, double-high, double-sized, or quad-sized characters
- utility graphics
- process graphics symbols
- reverse video (2000, 2005, 2060)
- high intensity (2000, 2005, 2060)

Attributes are assigned by sending an attribute command to the workstation immediately before the character string that is to be displayed. All characters subsequently received by the workstation will be displayed with that attribute until the assigned attribute is changed by sending a different attribute command to the workstation. Chapter 6 contains all command information.

5.6 CHARACTER SIZE CONSIDERATIONS

Each workstation is capable of displaying five sizes of characters: regular-size, double-wide, double-high, double-size, and quad-size. Different size characters can be shown on the video display simultaneously.

The relative sizes of the field for the different sizes of characters are shown in Figure 5-1 on the following page. Note that larger characters occupy fields that are multiples of the regular-size character field.

Care must be exercised in positioning the cursor when using the larger characters. This is because, in general, the cursor moves a single regular-size character field at a time. The exception to this rule is when a character is being written to the video display. In this case the cursor will advance the proper number of regular-size character fields automatically after the character is displayed, so that it is ready to accept another character of the same size. In addition, the cursor will do a carriage return followed by the proper number of linefeeds to start a new line if a character is received when the cursor is in the last column of a line. The cursor is also sensitive to character size when a linefeed, backspace, or carriage return is received. When cursor movement other than a linefeed, backspace, or carriage return is attempted within a large character field, the cursor may disappear. Cursor movement is explained on the following pages.

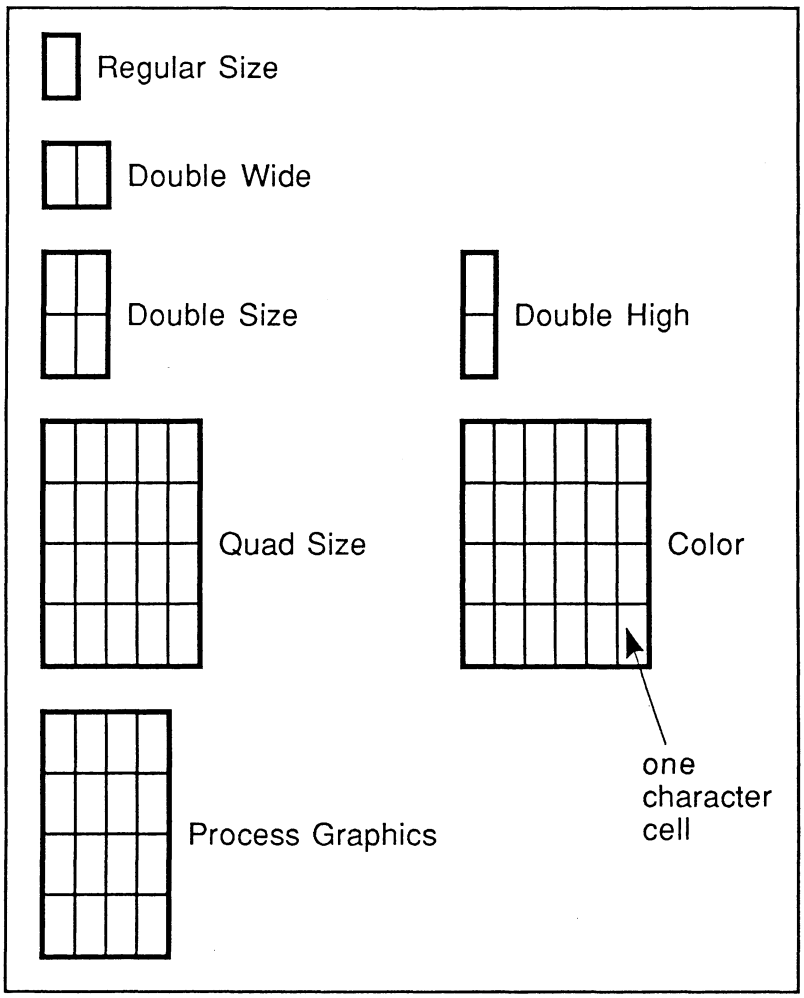
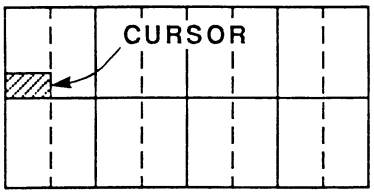
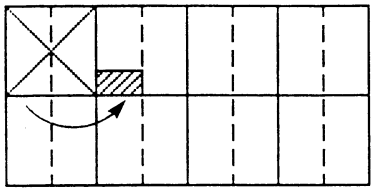


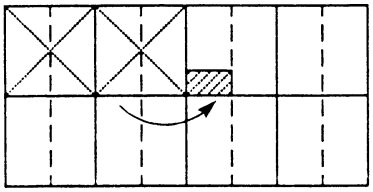
Figure 5-1. Relative Sizes of Character Fields



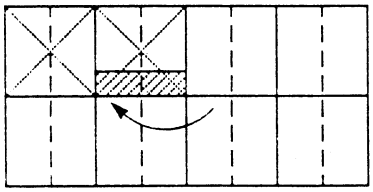
a) Character Size is Double-Wide with underline cursor positioned as shown. Note that cursor underlines only half of the double-wide character field.



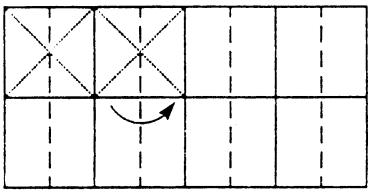
b) The character X is received and displayed and the cursor automatically advances two regular-size character fields to the next double-wide character field.



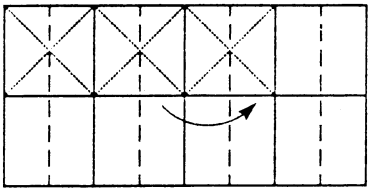
c) Another character X is received and displayed and the cursor advances again.



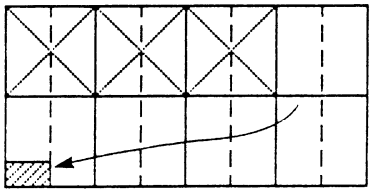
d) A single backspace command has been received and executed so the cursor is "pointing" to the left-hand regular-size character field of the second X, but now the cursor appears double-wide beneath the second X.



c) If a cursor right command is received and executed, the cursor moves to the position indicated and disappears.



f) If another X were received with the cursor located as in (e), the X would be displayed and the cursor would still be invisible.



g) If a carriage return and a linefeed or only a carriage return with automatic linefeed enabled is received, the cursor moves to the start of the first double-wide character field on the next line.

a) Character Size is Double-High with underline cursor positioned as shown. Note that the cursor is in the middle of the first double-high character field.

b) The character X is received and displayed and the cursor automatically advances one regular-size character field to the middle of the next double-high character field.

c) Another character X is received and displayed and the cursor advances again.

d) Here, a single backspace command has been received and executed and the cursor moves left one regular-size character field to the middle of the second X.

e) If a cursor right command is received and executed, the cursor moves back to the position it was at in (c).

f) Another character X is received and displayed and the cursor advances.

g) If a carriage return and a linefeed, or only a carriage return with automatic linefeed enabled is received, the cursor moves to the next line and is in the proper position to receive another character.

h) Another character X is received and displayed, and the cursor advances.

See Figure 5-2 below.

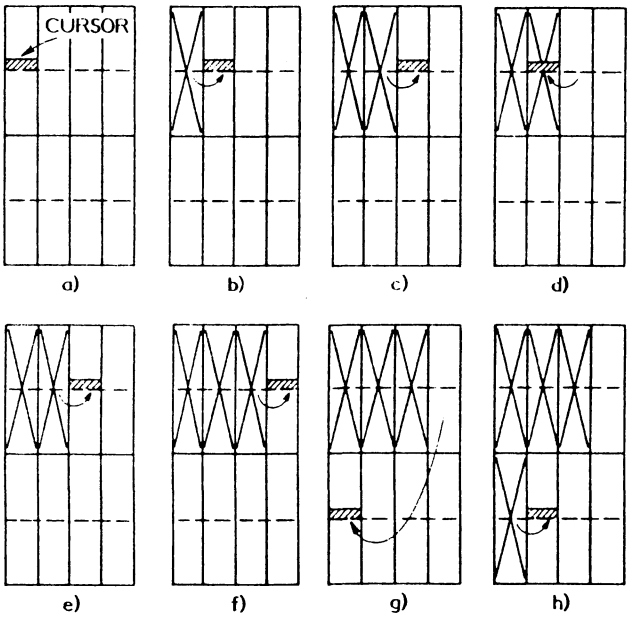


Figure 5-2. DOUBLE HIGH Character Cursor Movement

a) Character size is Double-High/Double-Wide with underline cursor positioned as shown. The cursor is in the middle of the double-high/double-wide character field and underlines only the first regular-size character field.

b) The character X is received and displayed and the cursor automatically advances two regular-size character fields to the middle of the next double-high/double-wide character field.

c) Another character X is received and displayed and the cursor advances again.

d) Single backspace command has been received and executed so the cursor is "pointing" to the upper left-hand regular-size character field of the second X.

e) If a cursor right command is received and executed and the cursor moves to the position indicated at (e) and disappears.

f) If another X were received with the cursor located as in (e), the X would be displayed and the cursor would still be invisible.

g) If a carriage return and a linefeed or only a carriage return with automatic linefeed enabled is received, the cursor moves to the proper position to write the next row of double-high/double-wide characters. In this position, the cursor underlines only the upper left-hand regular-size character field.

See Figure 5-3 below.

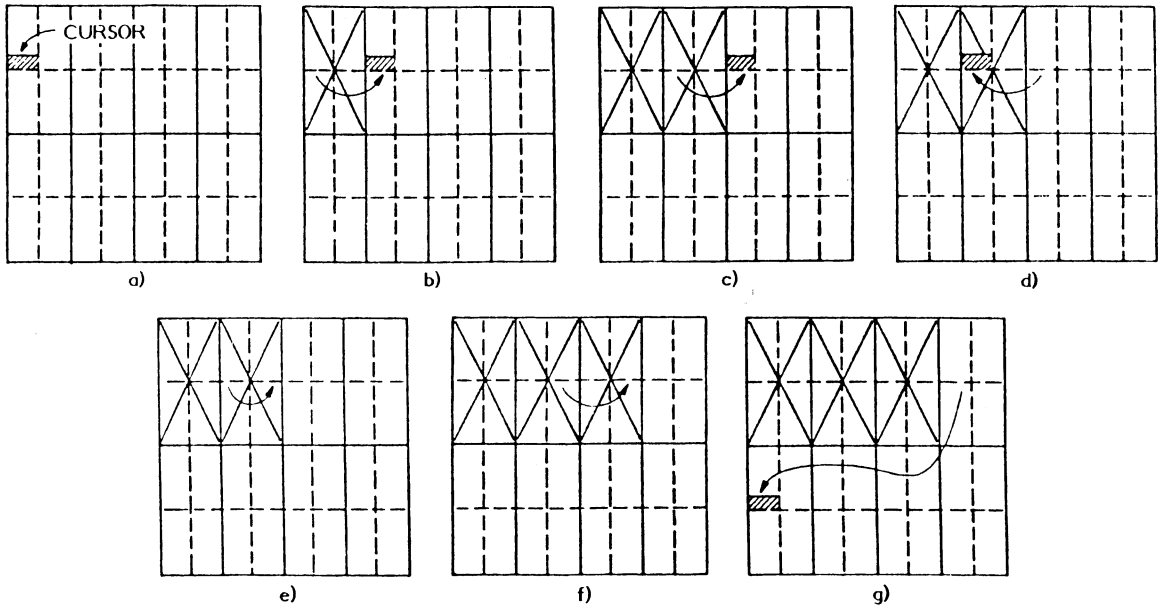


Figure 5-3. DOUBLE HIGH/ WIDE Character Cursor Movement

a) Character size is Quad-Size with underline cursor starting in position 1.

The character X is received and displayed and the cursor automatically advances to the start of the next quad-size character at 2. To move the cursor back to position 1 requires one backspace command. To move the cursor from 1 to 2 without writing a character requires five cursor right commands.

When moving within a quad-size character field, the cursor is always visible and remains a regular-size character.

b) To move the cursor from position 2 to the start of the first quad-size character field at 3, type a carriage return and linefeed, or only a carriage return with automatic linefeed enabled will move the cursor to 3.

If another displayable character code were received with the cursor within a displayed quad-size character, the new character would overwrite all or a portion of the existing character depending on the position of the cursor.

See Figure 5-4, on the following page.

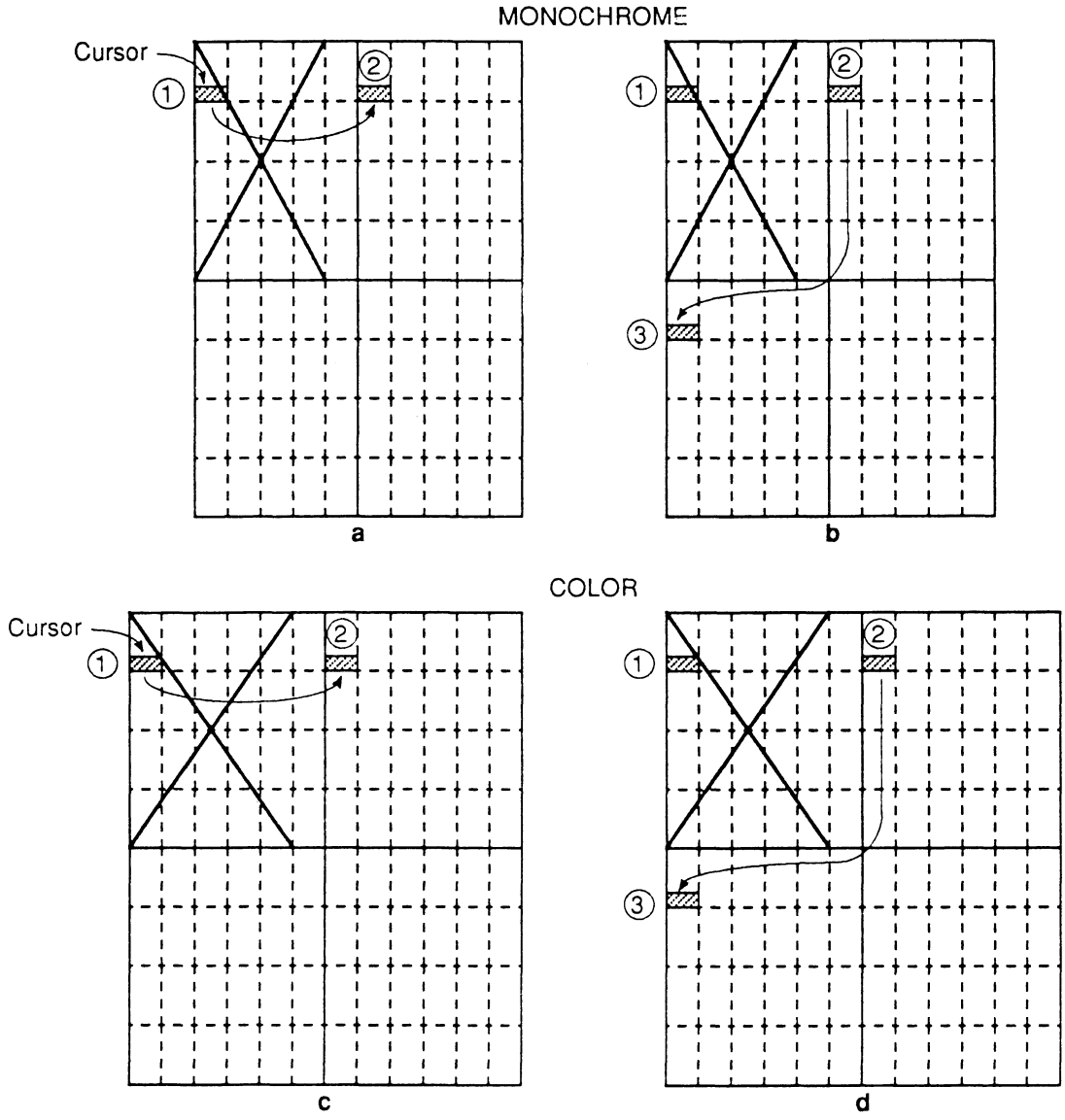


Figure 5-4. QUAD SIZE Character Cursor Movement

5.7 THIN-LINE GRAPHICS

Each workstation can display 16 different thin-line graphics characters (regular size and double wide only). These characters can be used, for example, to display diagrams on the video display. To display thin-line graphics characters, character set bits 0-2 of attribute byte No. 2 must be set to 000 (specifying regular characters).

The diagrams in Appendix B, character codes 128-143 decimal (80-8F hexadecimal), show the appearance of the 16 thin-line graphics characters.

> **NOTE**
> The workstation must be configured for 8 bits per character in order to display these codes via the serial port.

5.8 BLOCK GRAPHICS

The workstation can display 64 different regular-size and double wide block graphics characters, (16 for the 2050). These characters can be used in combination, for example, to display diagrams and characters that are larger than quad-size. Appendix B contains character codes 144-207 decimal (90-CF hexadecimal), (144-159 DEC, 90-9F hexadecimal for 2050) and shows all the block graphics characters.

Each block graphics character is made up of pixels. The different block graphics characters are made up by turning on different combinations of these pixels.

> **NOTE**
> The workstation must be configured for 8 bits per character in order for these characters to be displayed via codes from the serial port.

5.9 SPECIAL BAR GRAPHICS CHARACTERS

Special graphic characters are provided to draw solid character cells of specified heights and widths. These special bar graphics characters are regular or double size bits 0-2 of attribute byte No. 2 must be set to 00). There are four types of special bar graphics characters:

- Vertical bar up characters, which draw bars of varying heights, all beginning at the bottom of the character cell. See Appendix B, characters codes 209 - 222

decimal (D1 - DE hexadecimal).

- Vertical bar down characters, which begin at the top of the character cell and extend downwards. See Appendix B, character codes 225 - 238 decimal (E1 - EE hexadecimal).

- Horizontal bar right characters, which begin at the left edge of the character cell and extend to the right. See Appendix B, character codes 240 - 247 decimal (F0 - F7 hexadecimal).

- Horizontal bar left characters, which begin at the right edge of the character cell and extend to the left. See Appendix B, character codes 248 - 255 decimal (F8 - FF hexadecimal).

5.10 PROCESS GRAPHIC SYMBOLS

If the Character Set bits 0-2 of attribute byte No. 2 are zero when a displayable character is typed, the character printed on the key is displayed on the workstation screen. (These bits are automatically set to the default value 0 whenever the workstation is powered up or reset.) However, process graphic symbols can be selected by setting the Character Set bits to the following value:

Attribute Byte No. 2

Bit 2	Bit 1	Bit 0	Attribute
0	1	1	process graphic symbols

If process graphic symbols are selected, the character **transmitted** by the workstation will not change. However, certain characters codes sent to the display will cause graphic symbols to be displayed. For example, in process graphics mode, typing an uppercase "M" will still cause the character 4DH (hex value of "M") to be transmitted, but if an "M" is received, a small box instead of "M" will be displayed.

Table 5-2, on the following page, lists all the process graphic symbols and the characters which will generate each one.

NOTE
Graphic Presentation can be found in Appendix B, Table B-3.

Table 5-2. Process Graphic Symbols

Hex Value	ASCII Character	Process Control Symbol
20H		4x4 space
21H	!	motor in 4x3 cell
22H	"	not used
23H	#	left tank top in 4x1 cell
24H	$	right tank top in 4x1 cell
25H	%	small diamond in 4x2 cell
26H	&	left tank bottom in 4x1 cell
27H	'	right tank bottom in 4x1 cell
28H	(	left arrow in 4x2 cell
29H	)	right arrow in 4x2 cell
2AH	*	small box in 4x2 cell
2BH	+	up valve in 4x2 cell
2CH	,	right/left facing valve in 4x2 cell
2DH	-	pump/compressor in 4x2 cell
2EH	.	up arrow in 4x2 cell
2FH	/	down arrow in 4x2 cell
30H	0	small circle in 4x2 cell
31H	1	circuit breaker type 1 in 2x4 cell
32H	2	fuse in 2x4 cell
33H	3	disconnect in 3x4 cell
34H	4	pump/blower in 4x2 cell
35H	5	circuit breaker type 2 in 4x2 cell
36H	6	left turbine in 3x2 cell
37H	7	right turbine in 3x2 cell
38H	8	left medium box in 4x2 cell
39H	9	right medium box in 4x2 cell
3AH	;	left medium circle in 4x3 cell
3BH	:	right medium circle in 4x3 cell
3CH	<	mini circle in 2x1 cell
3DH	=	mini left arrow in 2x1 cell
3EH	>	mini right arrow in 2x1 cell
3FH	?	mini up arrow in 2x1 cell
40H	@	mini down arrow in 2x1 cell
41H	A	motor

Table 5-2. Process Graphic Symbols (cont.)

Hex Value	ASCII Character	Process Control Symbol
42H	B	large circle (left)
43H	C	large circle (right)
44H	D	tank top (left)
45H	E	tank top (right)
46H	F	small diamond
47H	G	large diamond (left)
48H	H	large diamond (right)
49H	I	tank bottom (left)
4AH	J	tank bottom (right)
4BH	K	left arrow
4CH	L	right arrow
4DH	M	small box
4EH	N	up facing valve
4FH	O	right/left facing valve
50H	P	pump/compressor
51H	Q	up arrow
52H	R	down arrow
53H	S	small circle
54H	T	transformer
55H	U	circuit breaker (type 1)
56H	V	fuse
57H	W	disconnect
58H	X	pump/blower
59H	Y	circuit breaker (type 2)
5AH	Z	turbine (left)
5BH	[	turbine (right)
5CH	\	large box (left)
5DH	]	large box (right)
5EH	^	medium box (left)
5FH	_ (underscore)	medium box (right)
60H	` (grave)	medium circle (left)
61H	a	medium circle (right)
62H	b	top left 1/4 of large circle in 4x2 cell
63H	c	top right 1/4 of large circle in 4x2 cell
64H	d	bottom left 1/4 of large circle in 4x2 cell
65H	e	bottom right 1/4 of large circle in 4x2 cell
66H	f	top left 1/4 of small circle in 2x1 cell
67H	g	top right 1/4 of small circle in 2x1 cell
68H	h	bottom left 1/4 of small circle in 2x1 cell
69H	i	bottom right 1/4 of small circle in 2x1 cell
6AH	j	small tank top in 4x1 cell

Table 5-2. Process Graphic Symbols (cont.)

Hex Value	ASCII Character	Process Control Symbol
6BH	k	small tank bottom in 4x1 cell
6CH	l	mini tank top in 2x1 cell
6DH	m	mini tank bottom in 2x1 cell
6EH	n	mini diamond in 2x1 cell
6FH	o	mini box in 2x1 cell
70H	p	mini right valve in 2x1 cell
71H	q	mini up valve in 2x1 cell
72H	r	mini motor in 2x2 cell
73H	s	mini pump/blower in 2x1 cell
74H	t	mini transformer in 2x2 cell
75H	u	mini circuit breaker type 1 in 1x2 cell
76H	v	mini fuse in 1x2 cell
77H	w	mini disconnect in 1x2 cell
78H	x	mini blower/compressor in 2x1 cell
79H	y	mini circuit breaker type 2 in 2x1 cell
7AH	z	mini left turbine in 1x1 cell
7BH	(	mini right turbine in 1x1 cell

If the workstation is in process graphic mode and a character not in the above table is typed or received, nothing will be displayed.

The process graphic symbols are shown in Appendix B.

5.11 UTILITY GRAPHICS

If the character set bits (bits 0-2 of attribute byte No. 2) are set to the value 111, the workstation will be in utility graphics mode. In this mode, receiving certain alphabetic characters will cause pieces of process control symbols to be displayed. The workstation uses these pieces to construct the process graphics symbols. You may be able to use these pieces to construct your own graphics, or to connect process graphic characters.

Note that this mode affects only the character/symbol **displayed** when certain character codes are sent to the display. It does not change the character **transmitted** by the workstation when the key is pressed.

Table 5-3, on the following page, describes the utility graphics available (see Appendix B for the complete chart of graphics characters available).

Table 5-3. Utility Graphics

Utility Graphics	Description
32-79 (20-4F Hex)	Process Graphics Pieces
80-87 (50-57 Hex)	Process Graphic Connectors (Thin)
88-95 (58-5F Hex)	Process Graphic Connectors (Thick)
96-111 (60-6F Hex)	Thick Line Graphics
112-175 (70-AF Hex)	Process Graphic Pieces
176-187 (B0-BB Hex)	Miscellaneous Connectors

5.12 GRAPHIC SHADING CHARACTERS

The shading characters can be used to create varying shades as used in filled boxes.

Appendix B, character codes 221, 222 and 237, 238 decimal (DD, DE and ED, EE hexadecimal), shows the shading graphic symbols.

6.1 INTRODUCTION

Remote commands allow the workstation to be controlled by the host device. Remote commands require lead-in character(s) to be received by the workstation immediately before the command code is received. In Hazeltine 1500 emulation, the lead-in character is ~ (7EH), called a tilde. In ANSI emulation, the lead-in character is ESC (1BH), or the two-character sequence ESC [(1BH 5BH). The lead-in code does not affect the display when received by the workstation.

If the code following the lead-in character is not a valid command code requiring a lead-in character, both the lead-in character and the code that follows it will be ignored by the workstation.

> **CAUTION**
> Configuration changes performed with remote commands are not saved when the workstation is turned off or reset.

If a remote command has been issued to change the current configuration, the new configuration is lost on power-down or reset. To save the new configuration, enter the first configuration menu and exit it (it is not necessary to change any parameters in the configuration menus). This saves the new configuration in EEPROM, so that the new configuration is effective upon power-up or reset.

See Tables 6-1 & 6-2 for a list of remote commands the workstation can receive from a host device.

6.2 HAZELTINE 1500 EMULATION

For a detailed description of the available Remote Commands, see Section 6.4. Cursor addressing is described in Section 5.2.

In Table 6-1, parameters such as <attr-1> or <xstart> are single bytes in the range 00H through FFH.

Table 6-1. Remote Commands
(Hazeltine 1500 Emulation)

REMOTE COMMANDS	ASCII	HEX
Control Characters		
Bell	<BEL>	07
Backspace	<BS>	08
Cursor to Next Foreground Field	<HT>	09
Linefeed	<LF>	0A
Carriage Return	<CR>	0D
Configuration Commands		
Enable Application Mode	~ .	7E 2E
Disable Application Mode	~ /	7E 2F
Cursor Off	~<S0H>	7E 01
Cursor On	~<STX>	7E 02
Scrolling Off	~<BEL>	7E 07
Scrolling On	~<BS>	7E 08
Unlock Keyboard	~<ACK>	7E 06
Lock Keyboard	~<NAK>	7E 15
Enable Printer Port	~ *	7E 2A
Disable Printer Port	~ +	7E 2B
Enable Screen Display	~ (	7E 28
Disable Screen Display	~)	7E 29
Attribute Commands		
Set/Reset Attributes	~6<attribute#>	7E 36 <attribute #>
Change Char. Attributes	~<ETX> <attr-1> <attr-2>	7E 03 <attr-1> <attr-2>
Cursor Movement Commands		
Cursor Right (no scroll)	<DLE>	10
Return Cursor Position	~<ENQ>	7E 05
Cursor Down (no scroll)	~<VT>	7E 0B
Cursor Up	~<FF>	7E 0C
Cursor to X,Y	~<DC1> X Y	7E 11 X Y
Home Cursor	~<DC2>	7E 12

Table 6-1. Remote Commands (cont.)
(Hazeltine 1500 Emulation)

REMOTE COMMANDS	ASCII	HEX
Clear Commands		
Clear to EOL with Background Spaces	~<SI>	7E 0F
Clear to EOS with Background Spaces	~<ETB>	7E 17
Clear to EOS with Foreground Spaces	~<CAN>	7E 18
Clear Foreground	~<GS>	7E 1D
Clear Screen	~<FS>	7E 1C
Background Field Follows	~<EM>	7E 19
Foreground Field Follows	~<US>	7E 1F
Delete Commands		
Delete Line	~<DC3>	7E 13
Insert Line	~<SUB>	7E 1A
Draw Commands		
Draw Box	~<HT> <char> <xstart> <ystart> <xend> <yend>	7E 09 <char> <xstart> <ystart> <xend> <yend>
Draw Vertical Line (upward)	~<LF> <char> <xstart> <ystart> <length>	7E 0A <char> <xstart> <ystart> <length>
Draw Horizontal Line (left to right)	~<CR> <char> <xstart> <ystart> <length>	7E 0D <char> <xstart> <ystart> <length>
Draw Bar Chart	~<S0> <xstart> <ystart> <length1> <length2>	7E 0E <xstart> <ystart> <length1> <length2>
Draw Bar Chart Down	~<space> <xstart> <ystart> <length1> <length2>	7E 20 <xstart> <ystart> <length1> <length2>
Draw Bar Chart Right	~! <xstart> <ystart> <length1> <length2>	7E 21 <xstart> <ystart> <length1> <length2>
Draw Bar Chart Left	~" <xstart> <ystart> <length1> <length2>	7E 22 <xstart> <ystart> <length1> <length2>

Table 6-1. Remote Commands (cont.)
(Hazeltine 1500 Emulation)

REMOTE COMMANDS	ASCII	HEX
Screen Control Commands		
Execute Stored Screen	~<DLE> <screen #>	7E 10 <screen #>
Receive Stored Screen	~<RS> <screen #>	7E 1E <screen #>
Transmit Stored Screen	~<ESC> <screen #>	7E 1B <screen #>
Jump to Stored Screen	~$ <screen #>	7E 24 <screen #>
Copy Screen Program	~, <source #><dest. #>	7E 2C <source #><dest. #>
Touch Screen Commands [1]		
Change to Normal Mode	~ : <NUL>	7E 3A 0
Change to Touch Screen Mode	~ : <SOH>	7E 3A 1
Set Programmable Touch Screen Zone	~ 9 <zone> <screen>	7E 39 <zone> <screen>
Define Zone or Zones	~ ; <upper left zone> <lower right zone> <code>	7E 3B <upper left zone> <lower right zone> <code>
Additional Commands		
Pause	~ # <time>	7E 23 <time>
Return Password	~ %	7E 25
Plot Point	~0XY	7E 30 X Y
Unplot Point	~1XY	7E 31 X Y

NOTES:

1 For more information on touch screen, see Section 4.2.

6.3 ANSI EMULATION

In ANSI mode, the parameters are one or more ASCII characters. Most parameters are numbers, with characters in the range 30H (the character "0") through 39H (the character "9").

In ANSI emulation, if the decimal value of a numeric parameter is greater than 9, two characters are necessary. For example, the decimal number 10 is represented as "1" followed by "0" (hex value 31,30). Likewise, if the decimal value is greater than 99, three characters are necessary.

ANSI values must be between 0-255.

In ANSI mode, parameters are separated by a semicolon, and all characters except <ESC> are displayable ASCII decimal characters.

Table 6-2. Remote Commands
(ANSI Emulation)

Control Characters

00	- ignored
07	- ring bell
08	- move cursor left 1 position
09	- go to next tab stop
0A	- linefeed or new line
0B	- same as 0A
0C	- same as 0A
0D	- move cursor to left margin of current line (carriage return)
18	- cancel current ESC sequence
1A	- same as 18
1B	- ESC

Configuration Commands [2,3]

ESC [? 7 h	- enable autowrap
ESC [? 25 h	- cursor on
ESC [? 7 l	- disable autowrap
ESC [? 25 l	- cursor off
ESC [2 h	- lock keyboard
ESC [2 l	- unlock keyboard
ESC [20 h	- enable auto line-feed
ESC [20 l	- disable auto line-feed
ESC [= 1 h	- cursor on
ESC [= 2 h	- scrolling on
ESC [= 3 h	- treat tab as ANSI tab
ESC [= 1 l	- cursor off
ESC [= 2 l	- scrolling off
ESC [= 3 l	- treat tab as Hazeltine tab
ESC [= 5 h	- enable printer port
ESC [= 5 l	- disable printer port
ESC [= 4 h	- enable screen display
ESC [= 4 l	- disable screen display

Table 6-2. Remote Commands (cont.)
(ANSI Emulation)

Attribute Commands [1]

ESC [1 ;attr1;attr2 p	- change character attributes
ESC [m	- attributes off
ESC [0 m	- attributes off
ESC [1 m	- highlight on
ESC [4 m	- underline
ESC [5 m	- blink
ESC [7 m	- reverse video
ESC [22 m	- highlight off
ESC [24 m	- underline disable
ESC [25 m	- blink disable
ESC [27 m	- reverse video off
ESC [30 m	- set character color to black
ESC [31 m	- set character color to blue
ESC [32 m	- set character color to green
ESC [33 m	- set character color to cyan
ESC [34 m	- set character color to red
ESC [35 m	- set character color to magenta
ESC [36 m	- set character color to yellow
ESC [37 m	- set character color to white
ESC [40 m	- set screen color to black
ESC [41 m	- set screen color to blue
ESC [42 m	- set screen color to green
ESC [43 m	- set screen color to cyan
ESC [44 m	- set screen color to red
ESC [45 m	- set screen color to magenta
ESC [46 m	- set screen color to yellow
ESC [47 m	- set screen color to white
ESC [50 m	- select regular character set
ESC [51 m	- select double-high characters
ESC [52 m	- select quad-sized characters
ESC [53 m	- select process control symbols
ESC [54 m	- select double-wide characters
ESC [55 m	- select double-size characters
ESC [56 m	- select quad-sized characters
ESC [57 m	- select utility graphics

Table 6-2. Remote Commands (cont.)
(ANSI Emulation)

Cursor Movement Commands

ESC [pn A	- cursor up pn lines
ESC [pn B	- cursor down pn lines without scroll
ESC [pn C	- cursor right pn characters
ESC [pn D	- cursor left pn characters
ESC [y;x H	- cursor to position x,y
ESC [H	- cursor home (1,1)
ESC [y;x f	- cursor to position x,y
ESC [f	- cursor home (1,1)
ESC D	- cursor down with scroll
ESC M	- cursor up with scroll
ESC E	- cursor to beginning of next line with scroll
ESC 7	- save cursor and attributes
ESC 8	- restore cursor and attributes

Tab Stop Commands [4]

ESC H	- set tab stop at current column
ESC [g	- clear tab stop at current column
ESC [0 g	- clear tab stop at current column
ESC [3 g	- clear all tab stops

Clear Commands

ESC [pn X	- clear pn characters on current line with background spaces
ESC [K	- clear to end of line with background spaces
ESC [? K	- clear to end of line with background spaces
ESC [0 K	- clear to end of line with background spaces
ESC [? 0 K	- clear to end of line with background spaces
ESC [1 K	- clear to beginning of line with background spaces
ESC [? 1 K	- clear to beginning of line with background spaces
ESC [2 K	- clear entire line with background spaces
ESC [? 2 K	- clear entire line with background spaces
ESC [J	- clear to end of screen with background spaces
ESC [? J	- clear to end of screen with background spaces
ESC [0 J	- clear to end of screen with background spaces
ESC [? 0 J	- clear to end of screen with background spaces
ESC [1 J	- clear to beginning of screen with background spaces
ESC [? 1 J	- clear to beginning of screen with background spaces
ESC [2 J	- clear entire screen with background spaces
ESC [? 2 J	- clear entire screen with background spaces

Table 6-2. Remote Commands (cont.)
(ANSI Emulation)

Insert/Delete Commands

ESC [pn L	- insert pn blank line(s) at current cursor position
ESC [pn M	- delete pn line(s) from cursor position
ESC [pn @	- insert pn space(s) in line at cursor position
ESC [pn P	- delete pn character(s) from line at cursor position

Report Commands

ESC [5 n	- device status report device ok returns - ESC [0 n device not ok returns - ESC [3 n
ESC [6 n	- report cursor x,y position, returns - ESC [y;xR
ESC [c	- return options
ESC [0 c	- return options, returns - ESC [? 1;0c

Screen Control Commands

ESC [12;<screen #>p	- execute stored screen
ESC [14;<screen #>p	- receive stored screen
ESC [13;<screen #>p	- transmit stored screen
ESC [19;<screen #>p	- jump to stored screen
ESC [23;<source #>;<dest. #>p	- copy screen program
ESC [24;<K>;<P1>p	- macro key command

Touch Screen Commands [5]

ESC [41; 0 p	- put workstation in normal mode
ESC [41; 1 p	- put workstation in touch screen mode
ESC [40;<zone>;<screen> p	- set programmable touch screen zone
ESC [42; <upper left zone>; <lower right zone>;<code> p	- define zone or zones

Additional Commands

ESC c	- reset to initial state
ESC b	- unlock keyboard
ESC ‘	- lock keyboard
ESC [18;time p	- pause
ESC [20 p	- return password

Table 6-2. Remote Commands (cont.)
(ANSI Emulation)

Draw Commands	
ESC [2 ;char;ystrt;xstrt;yend;xend p	- draw box
ESC [3 ;char;ystrt;xstrt;length p	- draw vertical line
ESC [4 ;char;ystrt;xstrt;length p	- draw horizontal line
ESC [5 ;ystrt;xstrt;len1;len2 p	- draw bar chart up
ESC [9 p	- background follows
ESC [10 p	- clear foreground
ESC [11 p	- foreground follows
ESC [15;ystrt;xstrt;len1;len2 p	- draw bar chart down
ESC [16;ystrt;xstrt;len1;len2 p	- draw bar chart right
ESC [17;ystrt;xstrt;len1;len2 p	- draw bar chart left
ESC [25;ycor;xcor p	- plot point
ESC [26;ycor;xcor p	- unplot point

NOTES:

1 Multiple attributes can be selected in a single attribute command:
ESC [50;40;31m

2 Multiple configurations can be specified in a single configuration command.
Example:
ESC [= 1;2;3 h
ESC [? 7;25 h
ESC [2;20 h

3 Configuration options that can be set by both the remote commands and the Configuration Menu are not saved on power-down unless the first configuration menu is entered and exited.

4 Tab stops set/reset with remote commands are not saved on power-down unless the Set Tab Stop menu is entered and exited.

5 For more information on touch screen, see Section 4.2.

6.3.1 VT100/220 Support

When the workstation is configured for ANSI mode, it emulates the DEC VT100 and VT220 terminals. Some VT100/220 commands, which are not handled by the workstation, are shown below. On the other hand, some ANSI commands are not supported by the VT100/220 and are available on the workstation.

The VT100/220 functions not emulated are listed below.

- 132 column mode is not supported
- not all special function keys are supported
- transmit and receive baud rates are not independent and there are fewer available baud rates
- no split screen capability
- different set-up procedure for configuration
- no user controllable LEDs
- no margin bell, key click
- optional DEC character sets and graphics are not supported
- VT52 mode is not supported
- can not invoke confidence tests remotely
- line attributes (double-high, double-wide) supported differently
- application mode supported only on the keypad, not on the keyboard
- insert mode not supported

When codes for these functions are received, they are ignored.

Appendix C lists the VT100/220 codes not supported by the workstation.

6.4 AVAILABLE REMOTE COMMANDS

Most of the remote commands listed in Table 6-1 are self-explanatory. However, some of the commands require further information, which is presented below, and some of the commands will affect the workstation's configuration options which are discussed in Chapter 3.

All commands may be entered either in hex or in ASCII, both for Hazeltine 1500 and ANSI emulation. However, hex is typically used in Hazeltine emulation, and ASCII is usual for ANSI emulation. Therefore, the remote commands for Hazeltine emulation are presented in hex in this chapter, while the remote commands for ANSI emulation are presented in ASCII.

Foreground and Background Fields

The following commands are related to the foreground and background fields on the workstation screen:

- Cursor to Next Foreground Field
- Clear to EOL with Background Spaces
- Clear to EOS with Background Spaces
- Clear to EOS with Foreground Spaces
- Background Field Follows
- Clear Foreground
- Foreground Field Follows
- Clear Screen
- Clear Line
- Clear to the Beginning of a Line
- Clear to the Beginning of a Screen
- Clear Characters on a Line

The workstation allows you to define foreground and background fields.

Foreground and background fields may be useful in distinguishing areas on the screen (e.g., column title fields).

Foreground fields are defined as characters whose **field** color is the same as the currently defined **character** color (character and field colors are defined by the Set Attribute command, see Section 5.5). All fields that are not foreground fields are background fields. Changing the character color with a Change Attribute command changes the definition of what fields are foreground fields.

6.4.1 Cursor to Next Foreground Field

Function: Moves the cursor to the first character in the next foreground field. In ANSI emulation, the TAB character (09H) defaults to "Cursor To Next Tab Stop." The ANSI user must issue ESC[=3l to use tabs to move the cursor to the next foreground field.

Hazeltine emulation: 09H
ANSI emulation: <ESC> [= 3 l (treat tab as Hazeltine tab (and the last character is a lower case "L"))
<HT>

6.4.2 Clear to EOL with Background Spaces

Function: All characters from the current cursor position to the end of the line are cleared to spaces. In addition, all character positions from the current cursor position to end of line are defined as a background field.

Hazeltine emulation: 7EH 0FH
ANSI emulation: <ESC> [K

6.4.3 Clear to EOS with Background Spaces

Function: All characters from the current cursor position to the end of screen are cleared to spaces. In addition, all character positions from the current cursor position to end of the screen are defined as a background field.

Hazeltine emulation: 7EH 17H
ANSI emulation: <ESC> [J

6.4.4 Clear to EOS with Foreground Spaces

Function: All characters from the current cursor position to the end of the screen are cleared to spaces. In addition, all character positions from the current cursor position to the end of screen are defined as a foreground field.

Hazeltine emulation: 7EH 18H
ANSI emulation: <ESC> [8 p

6.4.5 **Background Field Follows**

Function: All subsequent data is displayed as a background field, until a Foreground Field Follows command is executed.

Hazeltine emulation: 7EH 19H
ANSI emulation: <ESC> [9 p

6.4.6 **Clear Foreground**

Function: All foreground fields on the entire screen are replaced by foreground spaces, and the cursor is moved to the first position of the first foreground field.

Hazeltine emulation: 7EH 1DH
ANSI emulation: <ESC> [10 p

6.4.7 **Foreground Field Follows**

Function: All subsequent data is displayed as a foreground field, until a Background Field Follows command is executed.

Hazeltine emulation: 7EH 1FH
ANSI emulation: <ESC> [11 p

6.4.8 Clear Screen with Background Spaces

Function: All characters and data are cleared from the display screen.

Hazeltine emulation: 7EH 1CH
ANSI emulation: <ESC> [2 J

6.4.9 Clear Line with Background Spaces

Function: All characters are cleared from the current line the cursor is on.

Hazeltine emulation: N/A
ANSI emulation: <ESC> [K

6.4.10 Clear to Beginning of the Line with Background Spaces

Function: All characters are cleared from the current cursor position to the beginning of the current line.

Hazeltine emulation: N/A
ANSI emulation: <ESC> [1 K

6.4.11 Clear to the Beginning of the Screen with Background Spaces

Function: All characters are cleared from the current cursor position to the beginning of the screen.

Hazeltine emulation: N/A
ANSI emulation: <ESC> [1 J

6.4.12 Clear Characters on a Line with Background Spaces

Function: All specified number of characters are cleared on the current line.

Hazeltine emulation: N/A
ANSI emulation: <ESC> [pn X

where: pn = the number of characters to be cleared

The following example first defines some background fields, leaving the foreground fields blank. Then it homes the cursor and proceeds to fill the previously defined foreground fields with data.

ANSI

Command	Comments
<ESC> [2 J	-- Clear Screen
Weld Station:	-- Message on Screen
<ESC> [11 p	-- Foreground Field Follows
<SPACE>	-- Blank character, required to establish the rest of the line as foreground field
<CR>	-- Carriage Return
<LF>	-- Linefeed
<ESC> [9 p	-- Background Field Follows
Status:	-- Message on Screen
<ESC>[11 p	-- Foreground Field Follows
<SPACE>	-- Blank character
<CR>	-- Carriage Return
<LF>	-- Linefeed
<ESC> [11 p	-- Foreground field follows
<SPACE>	-- Blank character
<CR>	-- Carriage Return
<LF>	-- Linefeed
<ESC> [H	-- Home Cursor
<ESC> [= 3l	-- Treat Tab as Hazeltine Tab
<TAB>	-- Cursor to Next Foreground Field
Carriage Assembly - Left	-- Message on Screen
<TAB>	-- Cursor to Next Foreground Field
Not Operational	-- Message on Screen
<TAB>	-- Cursor to Next Foreground Field
Overcurrent Detected	-- Message on Screen

Hazeltine 1500

Note that the hexadecimal representations of the ASCII characters are listed, not the ASCII characters themselves.

Command	Comments
7EH 1CH	-- Clear Screen
WELD STATION:	-- Message on Screen
7EH 1FH	-- Foreground Field Follows
<SPACE>	-- Blank character, required to establish rest of line as foreground field
0D 0AH	-- Carriage Return, Linefeed
7EH 19H	-- Background Field Follows
STATUS:	-- Message on Screen
7EH 1FH	-- Foreground Field Follows
<SPACE>	-- Blank character
<CR>, <LF>	-- Carriage Return, Linefeed
7EH 1FH	-- Foreground Field Follows
20H	-- Space
0D 0AH	-- Carriage Return, Linefeed
7EH 12H	-- Home Cursor
09H	-- Cursor to Next Foreground Field
Carriage Assembly - Left	-- Message on Screen
09H	-- Cursor to Next Foreground Field
Not Operational	-- Message on Screen
09H	-- Cursor to Next Foreground Field
Overcurrent Detected	-- Message on Screen

6.4.13 **Draw Box**

Function: Draws a box. The coordinates of the upper left and lower right corners are included in the character sequence.

Hazeltine emulation: 7EH 09H <char> <xstart> <ystart> <xend> <yend>
ANSI emulation: <ESC>[2;<char>;<ystart>;<xstart>;<yend>;<xend>p

where:

char -- Hazeltine emulation:

01H = thick-line box
02H = thin-line box
03H = thin-line box using utility graphics characters
04H = thick-line box using utility graphics characters
Any displayable ASCII character = box composed of that character.

-- ANSI emulation:

1(31H) = thick-line box
2(32H) = thin-line box
3(33H) = thin-line box using utility graphics characters
4(34H) = thick-line box using utility graphics characters
Sequence of two ASCII decimal characters = box composed of the ASCII equivalent of the decimal value. For example, to draw a box composed of the character "A"(65) the following two characters are required: 6(36H) and 5(35H).

xstart = x coordinate of upper left corner of box
ystart = y coordinate of upper left corner of box
xend = x coordinate of lower right corner of box
yend = y coordinate of lower right corner of box

NOTE
This command will not cause automatic scrolling if a box the size if the screen is drawn.

6.4.14 **Draw Vertical Line in Upward Direction**

Function: Draws an upward vertical line, beginning at the coordinate included in the command sequence, toward the screen's top edge.

Hazeltine emulation: 7EH 0AH <char> <xstart> <ystart> <length>
ANSI emulation: <ESC>[3;<char>;<ystart>;<xstart>;<length>p

where:

char -- Hazeltine emulation:

01H = thick line
02H = thin line
03H = thin right of cell connector (utility graphic 51H)
04H = thin left of cell connector (utility graphic 53H)
05H = thick right of cell connector (utility graphic 59H)
06H = thick left of cell connector (utility graphic 5BH)
Any displayable ASCII character = line composed of that character.

-- ANSI emulation:

1(31H) = thick line
2(32H) = thin line
3(33H) = thin right of cell connector (utility graphic 51H)
4(34H) = thin left of cell connector (utility graphic 53H)
5(35H) = thick right of cell connector (utility graphic 59H)
6(36H) = thick left of cell connector (utility graphic 5BH)
Sequence of two ASCII decimal characters = line composed of the ASCII equivalent of the decimal value. For example, to draw a line composed of the character "A"(65) the following two characters are required: 6(36H) and 5(35H).

xstart = x coordinate of start of line
ystart = y coordinate of start of line
length = length of line (in units of character cells)

6.4.15 **Draw Horizontal Line from Left to Right**

Function: Draws a horizontal line (from left to right) starting at the coordinate in the character sequence, toward the right edge of the screen.

Hazeltine emulation: 7EH 0DH <char> <xstart> <ystart> <length>
ANSI emulation: <ESC>[4;<char>;<ystart>;<xstart>;<length>p

where:

char -- Hazeltine emulation:

01H = thick line
02H = thin line
03H = thin top of cell connector (utility graphic 50H)
04H = thin bottom of cell connector (utility graphic 52H)
05H = thick top of cell connector (utility graphic 58H)
06H = thick bottom of cell connector (utility graphic 5AH)
Any displayable ASCII character = line composed of that character.

-- ANSI emulation:

1(31H) = thick line
2(32H) = thin line
3(33H) = thin top of cell connector (utility graphic 50H)
4(34H) = thin bottom of cell connector (utility graphic 52H)
5(35H) = thick top of cell connector (utility graphic 58H)
6(36H) = thick bottom of cell connector (utility graphic 5AH)
Sequence of two ASCII decimal characters = line composed of the ASCII equivalent of the decimal value. For example, to draw a line composed of the character "A"(65) the following two characters are required: 6(36H) and 5(35H).

xstart = x coordinate of start of line
ystart = y coordinate of start of line
length = length of line (in units of character cells)

6.4.16 **Draw Bar Up**

Function: Draws a high-resolution vertical bar one character wide. The coordinate of the bottom character cell of the bar and its height are included in the character sequence. This command includes a character specifying the height of the bar to be erased before the new bar is drawn, so that bars can be updated dynamically.

Hazeltine emulation: 7EH 0EH <xstart> <ystart> <length1> <length2>
ANSI emulation: <ESC>[5;<ystart>;<xstart>;<length1>;<length2>p

where: xstart is the x coordinate of start of bar
ystart is the y coordinate of start of bar
length1 is the height of column (in units of 1/12 of a character cell)
The height must be in the range 0 through 255. 12 is equivalent to the height of one character, 252 is equal to the height of 21 characters. For the 2050, (in units of 1/10), 10 is equivalent to the height of one character, 240 is equal to the height of 24 characters.
length2 is the height of the previous bar to be erased
Before the new vertical bar is drawn, a blank bar of length2 is drawn. This erases the previous bar. If length2 is zero, no blank line will be drawn.

6.4.17 **Draw Bar Down**

Function: Same as Draw Bar Up, except that bar is drawn downward, and <xstart> and <ystart> specify the top character cell of the bar.

Hazeltine emulation: 7EH 20H <xstart> <ystart> <length1> <length2>
ANSI emulation: <ESC>[15;<ystart>;<xstart>;<length1>;<length2>p

where: xstart is the x coordinate of start of bar
ystart is the y coordinate of start of bar
length1 is the height of column (in units of 1/12 of a character cell)
The height must be in the range 0 through 255. 12 is equivalent to the height of one character, 252 is equal to the height of 21 characters. For the 2050, (in units of 1/10), 10 is equivalent to the height of one character, 240 is equal to the height of 24 characters.
length2 is the height of the previous bar to be erased
Before the new vertical bar is drawn, a blank bar of length2 is drawn. This erases the previous bar. If length2 is zero, no blank line will be drawn.

6.4.18 **Draw Bar Right**

> **NOTE**
> For the Draw Bar Right and Left commands, multiple bars will be necessary to span up to the full width of the screen (80 characters).

Function: Same as Draw Bar Up, except that bar is drawn to the right, and <xstart> and <ystart> specify the left end character cell of the bar.

Hazeltine emulation: 7EH 21H <xstart> <ystart> <length1> <length2>
ANSI emulation: <ESC>[16;<ystart>;<xstart>;<length1>;<length2>p

where: xstart is the x coordinate of start of bar
ystart is the y coordinate of start of bar
length1 is the width of the bar (in units of 1/5 of a character cell)
The width must be in the range 0 through 255. 5 is equivalent to the width of one character, 255 is equal to the width of 51 characters. For the 2050, (in units of 1/8), 8 is equivalent to the width of one character, 248 is equal to the width of 31 characters.
length2 is the width of the previous bar to be erased
Before the new horizontal bar is drawn, a blank bar of length2 is drawn. This erases the previous bar. If length2 is zero, no blank line will be drawn.

6.4.19 **Draw Bar Left**

Function: Same as Draw Bar Right, except that bar is drawn to the left, and <xstart> and <ystart> specify the right end character cell of the bar.

Hazeltine emulation: 7EH 22H <xstart> <ystart> <length1> <length2>
ANSI emulation: <ESC>[17;<ystart>;<xstart>;<length1>;<length2>p

where: xstart is the x coordinate of start of bar
ystart is the y coordinate of start of bar
length1 is the width of the bar (in units of 1/5 of a character cell)
The width must be in the range 0 through 255. 5 is equivalent to the width of one character, 255 is equal to the width of 51 characters. For the 2050, (in units of 1/8), 8 is equivalent to the width of one character, 248 is equal to the width of 31 characters.
length2 is the width of the previous bar to be erased
Before the new horizontal bar is drawn, a blank bar of length2 is drawn. This erases the previous bar. If length2 is zero, no blank line will be drawn.

6.4.20 **Pause**

Function: Causes the workstation to pause for a specified period before retrieving and displaying the next character or command from the serial port or screen program.

Hazeltine emulation: 7EH 23H<time>
ANSI emulation: <ESC>[18;<time>p

where: time = duration of pause (in tenths of a second)

6.4.21 **Cursor Off**

> **NOTE**
> Cursor is always ON when the workstation enters Operating Mode (from set-up or power-up).

Function: Makes the cursor invisible.

Hazeltine emulation: 7EH 01H
ANSI emulation: <ESC>[=1l

6.4.22 **Cursor On**

Function: Makes the cursor visible.

Hazeltine emulation: 7EH 02H
ANSI emulation: <ESC>[=1h

6.4.23 **Scrolling Off**

> **NOTE**
> Scrolling can also be turned on or off via the Miscellaneous Configuraiton Menu (see Section 3.4.3). The value set with the Scrolling On or Scrolling Off remote commands is not saved at power-up or reset. The setting in the Miscellaneous Configuration Menu is used to set scrolling on/off after power-up or reset.

Function: Disables screen scrolling. Any keystroke or serial input that would normally cause the screen to scroll will instead cause the cursor to go to the top of the screen.

Hazeltine emulation: 7EH 07H
ANSI emulation: <ESC>[=2l

6.4.24 **Scrolling On**

Function: Enables screen scrolling. Used to re-enable scrolling after the Scrolling Off command has been used to disable scrolling.

Hazeltine emulation: 7EH 08H
ANSI emulation: <ESC>[=2h

6.4.25 **Insert Line**

Function: Inserts a line (or lines) immediately before the current line, and moves the cursor to the beginning of the inserted line.

Hazeltine emulation: 7EH 1AH
ANSI emulation: <ESC>[pn L

where: pn is the number of blank lines to insert.

6.4.26 **Delete Line**

Function: Deletes the line on which the cursor is positioned.

Hazeltine emulation: 7EH 13H
ANSI emulation: <ESC>[pn M

where: pn is the number of lines to delete

6.4.27 **Plot Point**

Function: Turns on one pixel. Each character cell consists of six pixels (four pixels for the 2050).

Hazeltine emulation: 7EH 30H <x><y>
ANSI emulation: <ESC>[25;<y>;<x>p

where: x is the horizontal coordinate (0-159)
y is the vertical coordinate (0-71) or (0-47) for the 2050

(Note that the lower left-hand corner has coordinates 0,0.)

6.4.28 **Unplot Point**

Function: Turns off one pixel. If the specified pixel is not on, this command has no effect.

Hazeltine emulation: 7EH 31H <x><y>
ANSI emulation: <ESC>[26;<ycor>;<xcor>p

where: xcor is the horizontal coordinate (0-159)
ycor is the vertical coordinate (0-71) or (0-47) for the 2050

(Note that the lower left-hand corner has coordinates 0,0.)

6.4.29 **Return Password**

Function: Returns the current password (three characters), followed by a carriage return. If the password is disabled (via option #3 of the Main Menu, see Section 3.6), only a carriage return is transmitted.

Hazeltine emulation: 7EH 25H
ANSI emulation: <ESC>[20p

6.4.30 **Set/Reset Attributes**

Function: Sets/resets the workstation attributes.

Hazeltine emulation: 7EH 36H xx
ANSI emulation: <ESC>[<dd> m

where: xx = the attribute set/reset hazeltine code:
dd = the attribute set/reset ANSI code:

dd	xx
0	(00H) - attributes off
1	(01H) - highlight on
4	(04H) - underscore on
5	(05H) - blink on
7	(07H) - reverse video on
22	(16H) - highlight off
24	(18H) - underscore off
25	(19H) - blink off
27	(1BH) - reverse video off
50	(32H) - select regular characters
51	(33H) - select double-high characters
52	(34H) - select quad-size characters
53	(35H) - select process graphics characters
54	(36H) - select double-wide characters
55	(37H) - select double-size characters
56	(38H) - select quad-size characters
57	(39H) - select utility graphics

The following page displays the 2050 attribute list.

2050 Attribute List

0 (00H) - attributes off (does not affect colors)
4 (04H) - underscore on
5 (05H) - blink on
7 (07H) - reverse colors
24 (18H) - underscore off
25 (19H) - blink off
30 (1EH) - set character color to black
31 (1FH) - set character color to blue
32 (20H) - set character color to green
33 (21H) - set character color to cyan
34 (22H) - set character color to red
35 (23H) - set character color to magenta
36 (24H) - set character color to yellow
37 (25H) - set character color to white
40 (28H) - set screen color to black
41 (29H) - set screen color to blue
42 (2AH) - set screen color to green
43 (2BH) - set screen color to cyan
44 (2CH) - set screen color to red
45 (2DH) - set screen color to magenta
46 (2EH) - set screen color to yellow
47 (2FH) - set screen color to white
50 (32H) - select regular characters
51 (33H) - select double-high characters
52 (34H) - select quad-size characters
53 (35H) - select process graphics characters
54 (36H) - select double-wide characters
55 (37H) - select double-size characters
56 (38H) - select quad-size characters
57 (39H) - select utility graphics

6.4.31 **Return Cursor Position**

Function: To read the cursor position, transmit a Return Cursor Position command to the workstation:

Hazeltine emulation: 7EH 05H
ANSI emulation: <ESC>[6 n

Hazeltine:

The workstation will then transmit the response:

<column coordinate> <row coordinate> CR

where: <column coordinate> will be a hex value between 20H-4FH and 60H-7FH, while <row coordinate> will be a hex value between 60H and 78H. CR is the ASCII character corresponding to 0D (hex).

Figure 6-2 lists the row and column coordinates under Hazeltine emulation.

ANSI:

The workstation will then transmit the response:

ESC [<row>;<column>R (ASCII character)

where: the row and column are ASCII decimal values (hex values between 30H and 39H). For example, if the cursor is currently in row 12, column 6, the Return Cursor Position command will return the following sequence of ASCII characters:

ESC [12; 06 R (ASCII character)

6.4.32 **Cursor to X,Y**

Function: To move the cursor to column x, row y on the screen.

Hazeltine emulation: 7EH 11H <x> <y>
ANSI emulation: <ESC> [y;x H

where: y = row
x = column

Hazeltine:

To move the cursor to column x, row y, transmit a Cursor to X,Y command to the workstation:

7EH 11H xx yy

where: xx and yy are the hexadecimal equivalents of the decimal values x,y (e.g., position 19=13 hex), and <char x> and <char y> are the ASCII characters corresponding to the hex values xx and yy (e.g., ASCII DC3 corresponds to 13 hex).

Figure 6-2 lists the row and column coordinates under Hazeltine emulation.

ANSI:

To move the cursor to row y, column x, transmit a Cursor to X,Y command to the workstation:

ESC [y;x H

> **NOTE**
> The decimal coordinate greater than 9 must be expressed as **two** decimal ASCII characters. For example, decimal coordinate 10 is expressed as "1" followed by "0" (31H 30H).

Figure 6-1 lists the row and column coordinates under ANSI emulation.

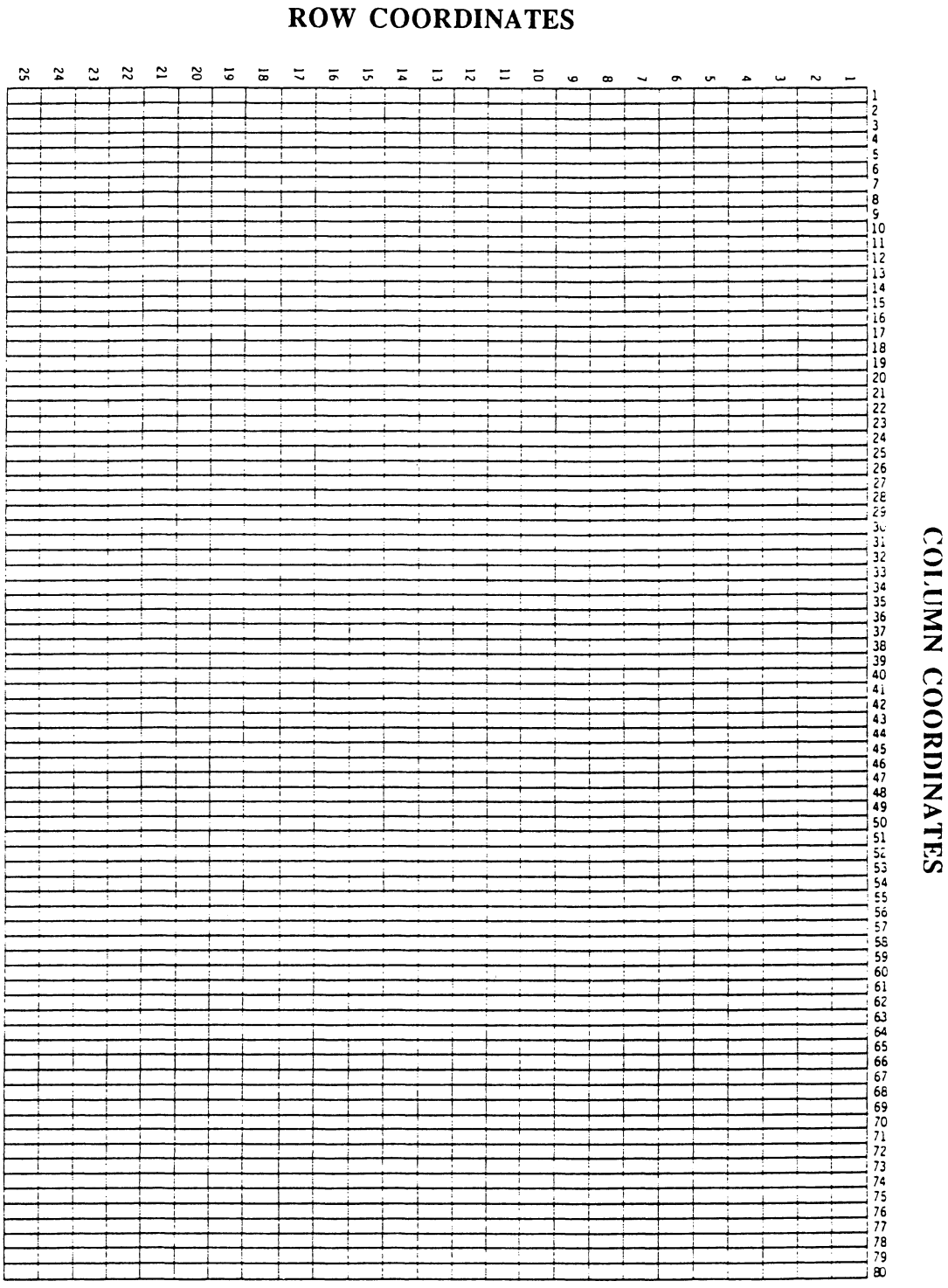


Figure 6-1. Video Display Coordinate System (ANSI Emulation)

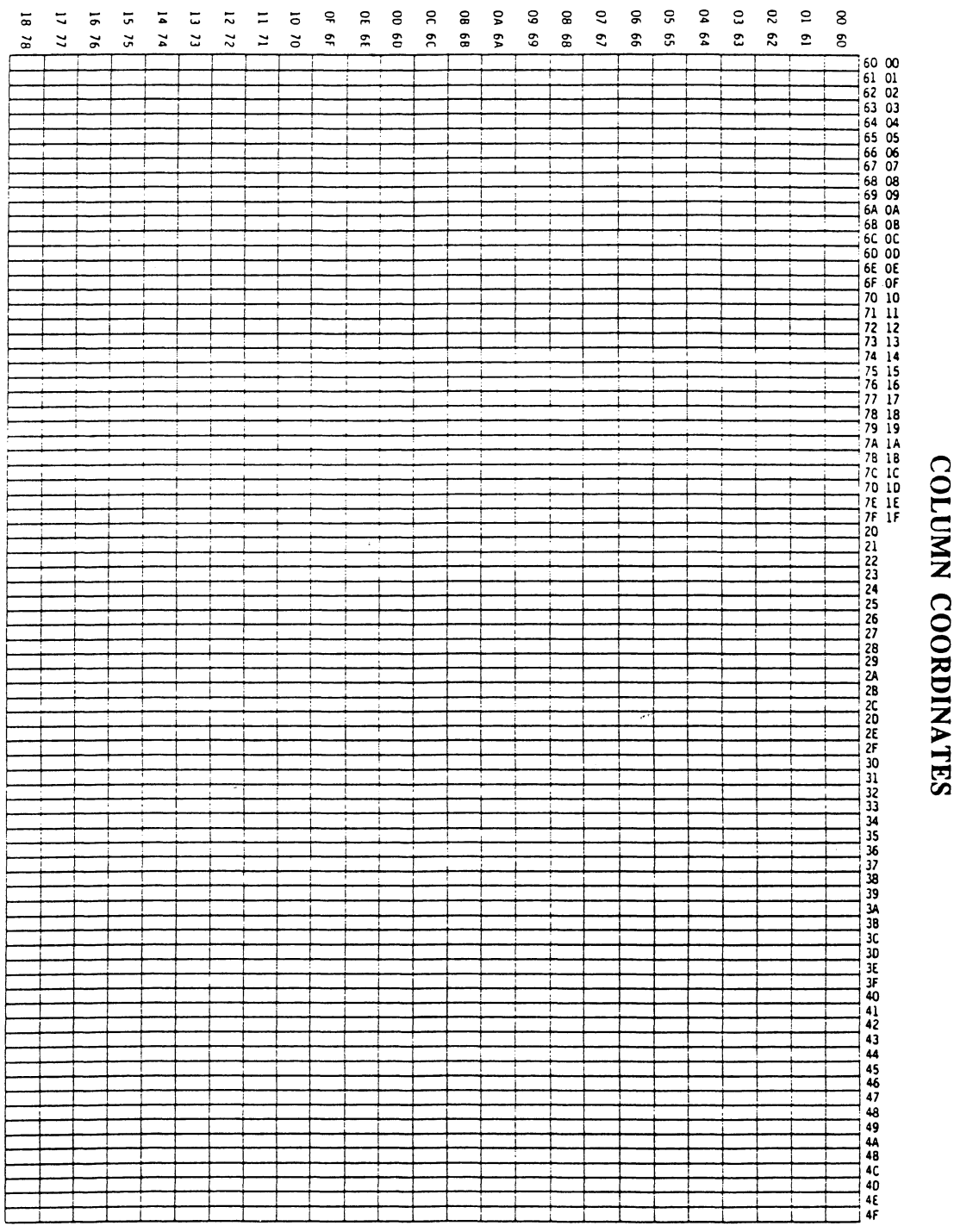


Figure 6-2. Video Display Coordinate System (Hazeltine Emulation)

6.4.33 **Change Character Attributes**

Function: This command is used to change a character's attribute.

Hazeltine emulation: 7EH 03H <attribute byte 1> <attribute byte 2>
ANSI emulation: <ESC> [1;<attribute byte 1>; <attribute byte 2> p

The definition of attribute byte 1 is shown in Table 6-3.

Table 6-3. Attribute Byte 1

Bit Number	Attribute
7 (MSB)	not used
6	not used character color bit 2 (2050)
5	not used character color bit 1 (2050)
4	double wide character color bit 0 (2050)
3	blink
2	underline
1	highlight double-wide (2050)
0 (LSB)	reverse video not used (2050)

The definition of attribute byte No. 2 is shown on the following page in Table 6-4.

Table 6-4. Attribute Byte 2

Bit Number	Attribute
7 (MSB)	not used
6	not used character field color bit 2 (2050)
5	not used character field color bit 1 (2050)
4	not used character field color bit 0 (2050)
3	not used
2	character set bit 2
1	character set bit 1
0 (LSB)	character set bit 0

The settings of bits 2 to 0 of attribute byte No. 2 can be any of the following (see Table 6-5):

Bit 2	Bit 1	Bit 0	Attribute
0	0	0	regular character
0	0	1	double-high character
0	1	0	quad-size character
0	1	1	process graphic symbols
1	1	1	utility graphics

Bits 6 to 4 of attribute byte No. 1 control the color of the **character** (i.e., foreground), while bits 6 to 4 of the attribute byte No. 2 control the color of the **field** (i.e., background) in which the character (or graphic symbol) is situated. For example, if the character color is red and the field color is yellow, red characters will be displayed on a yellow background.

The colors of both character and character field are defined in the same way, as shown in Table 6-5.

Table 6-5. Color Select Bits Attribute Byte 1 and 2

Bit 6[1] (red)	Bit 5[1] (green)	Bit 4[1] (blue)	Color
0	0	0	Black
0	0	1	Blue
0	1	0	Green
0	1	1	Cyan
1	0	0	Red
1	0	1	Magenta
1	1	0	Yellow
1	1	1	White

1 In attribute byte 1, these bits select the character color. In attribute byte 2, these bits select the character field color.

NOTE

The field color of any character displayed on the screen is independently programmable. Also, altering the character and character field colors affects only characters/symbols to be displayed, not character/symbols already displayed on the screen.

On power-up, the attributes are white characters, blue screen, regular character set, no underline, no blink.

6.4.34 Insert Spaces in a Line

Function: This command inserts spaces into a line beginning at the cursor's current position. Any characters from the cursor's current position to the end of the line will be removed.

Hazeltine emulation: N/A
ANSI emulation: <ESC> [pn @

where: pn = the number of spaces to be inserted

6.4.35 Delete Characters in a Line

Function: This command deletes characters from a line at the cursor's current position and inserts spaces at the end of the line.

Hazeltine emulation: N/A
ANSI emulation: <ESC> [pn P

where: pn = the number of characters to be deleted

6.4.36 Saving Cursor Attributes

Function: This command saves the cursor's current position, character set, autowrap flag state, and all attributes.

Hazeltine emulation: N/A
ANSI emulation: <ESC> 7

6.4.37 Restoring Cursor Attributes

Function: This command restore's the current cursor position, character set, autowrap flag state, and all attributes.

Hazeltine emulation: N/A
ANSI emulation: <ESC> 8

6.4.38 **Enable Printer Port**

Function: Enables the port selected for printing via the Miscellaneous Configuration Menu (see Section 3.4.3) and causes any text sent to the terminal to be echoed to the port.

Hazeltine emulation: 7EH 2AH
ANSI emulation: <ESC> [=5h

6.4.39 **Disable Printer Port**

Function: Disables the port selected for printing via the Miscellaneous Configuration Menu (see Section 3.4.3).

Hazeltine emulation: 7EH 2BH
ANSI emulation: <ESC> [5l

The serial printer port output buffer is 64 bytes long. If the character rate of the printer is slower than the character rate of the workstation, the workstation must wait each time it has a character to send to the printer. While it is waiting, the workstation can not display any other characters or execute any commands coming in over the serial port. Thus, if handshaking is not used, the workstation's input buffer could fill up and data would be lost.

The enable/disable printer port commands can be used in conjunction with the enable/disable screen display commands for a variety of screen/printer port combinations. These remote commands can be used to cause text to appear on both the screen and the printer port, or on just the screen, or on just the printer port.

6.4.40 **Enable Screen**

Function: Enables the screen display.

> **NOTE**
> The screen display is enabled on power-up.

Hazeltine emulation: 7EH 28H
ANSI emulation: <ESC>[= 4h

6.4.41 **Disable Screen**

Function: Disables the screen display. Text destined for the screen will not be displayed.

Hazeltine emulation: 7EH 29H
ANSI emulation: <ESC>[= 4l

When the screen is enabled all remote commands are processed normally. When the screen is disabled all remote commands (except those to enable/disable the printer or screen) and all text will be ignored and will not affect the screen display.

When the screen and printer port are both enabled most control characters and text are sent to both the screen and the printer. Remote commands go only to the screen and are not sent to the printer. If sequences of control characters (which must be sent to the printer) contain any of the lead-in control characters for valid ANSI or Hazeltine commands (i.e., Tilde or <ESC> followed by valid characters), then the first one or two characters will be treated as remote commands and will not go to the printer. If the characters immediately following these first characters do not constitute a command, they will be passed "as is" to the printer port.

When the screen is disabled and the printer port is enabled, all control codes and text will be sent to the printer except for remote commands to enable/disable the screen or the printer port.

6.4.42 **Execute Stored Screen**

Function: Causes the workstation to execute the specified screen program. If this command is included in a screen program, after the specified program is executed, control is returned to the calling program. Screen program nesting can be up to 10 levels deep.

Hazeltine emulation: 7EH 10H <screen #>
ANSI emulation: <ESC>[12;<screen #>p

6.4.43 **Receive Stored Screen**

Function: Causes the workstation to accept and store a screen which must be imbedded within the command. This screen must be terminated by DEL (7FH).

Hazeltine emulation: 7EH 1EH <screen #>
ANSI emulation: <ESC>[14;<screen #>p

Example: To send screen 5:

In Hazeltine mode,
Host sends: 7EH 1EH 05H <text of stored screen #5> 7FH

In ANSI mode,
Host sends: <ESC>[14;5p <text of stored screen #5> <DEL>

6.4.44 Transmit Stored Screen

Function: Causes the workstation to transmit a Receive Stored Screen command, followed by the actual stored screen, followed by DEL (7FH).

Hazeltine emulation: 7EH 1BH <screen #>
ANSI emulation: <ESC>[13;<screen #>p

Example: To send screen 35:

In Hazeltine mode,
Host sends: 7EH 1BH 23H
Terminal responds: 7EH 1EH 23H <text for stored screen #35> <DEL>

In ANSI mode:
Host sends: <ESC>[13;35p
Terminal responds: <ESC>[14;35p <text for stored screen #35> <DEL>

6.4.45 Jump to Stored Screen

Function: Jumps to the specified stored screen and begins executing that screen. Unlike the Execute Stored Screen command, this command does not return to the calling screen.

Hazeltine emulation: 7EH 24H <screen #>
ANSI emulation: <ESC>[19;<screen #>p

6.4.46 Copy Screen Program

Function: Copies the contents of any screen program into another screen program.

Hazeltine emulation: 7EH 2CH <source #><destination #>
ANSI emulation: <ESC>[23;<source #>;<destination #>;p

6.4.47 **2005 Programmable Keypad Key**

Function: Associates a screen program with a key on the keypad. Thereafter, each time that key is pressed, the contents of that screen program are transmitted out the serial port. In this way you can set your own "Macro" key definitions.

When the terminal is in half-duplex and a keypad key associated with a program is pressed, the following happens:

1. The contents of the program is transmitted out the serial port.
2. The program is executed by the terminal and the results are displayed on the screen.

Hazeltine emulation: 7EH 2DH <K><P1>

ANSI emulation: <ESC>[24;<K>;<P1>p

where: <K> = Key number (0-53, see Figure 6-3, below)
<P1> = Screen number (1-255)

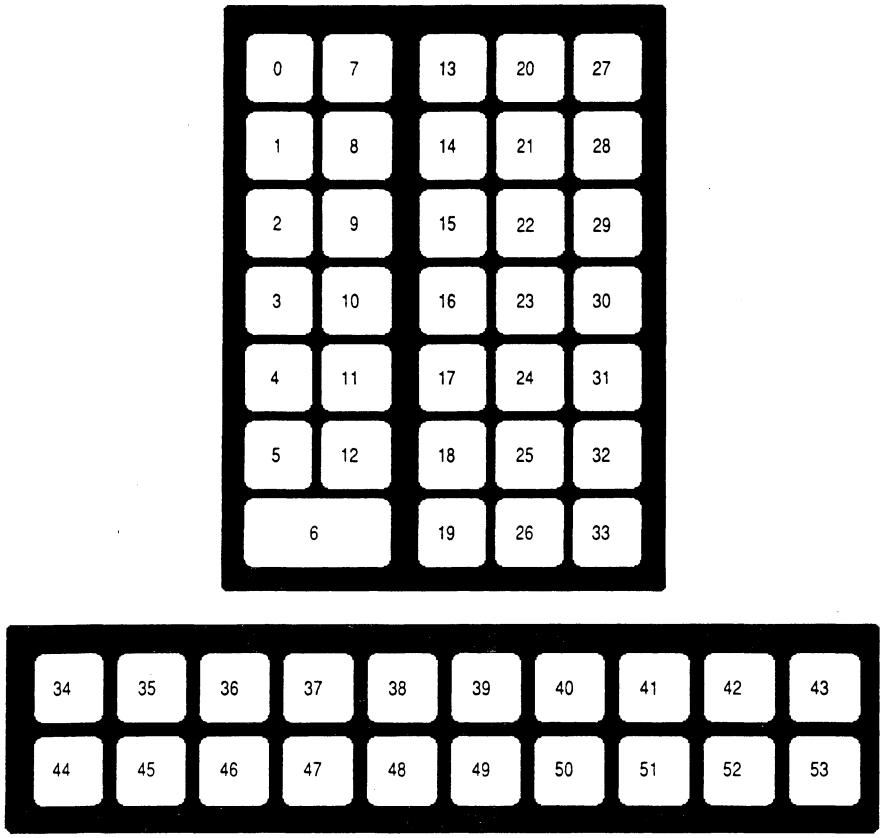


Figure 6-3. 2005 Programmable Keypad and Function Keys

6.4.48 2050/2060 Programmable Keypad Key

Function: Associates a screen with a key on the keypad. Thereafter, each time that the key is pressed, the contents of that screen program are transmitted out the serial port. In this way you can set your own "Macro" key definitions.

When the terminal is in half-duplex and a keypad key associated with a program is pressed, the following happens:

1. The contents of the program is transmitted out the serial port.
2. The program is executed by the terminal and the results are displayed on the screen.

Hazeltine emulation: 7EH 2DH <K><P1>

ANSI emulation: <ESC>[24;<K>;<P1>p

where:	<K>	= Key number (0-57, see Figure 6-4 on the following page)
	<P1>	= Screen number (1-255)
if:	<P1>	= 0, reset the key to none

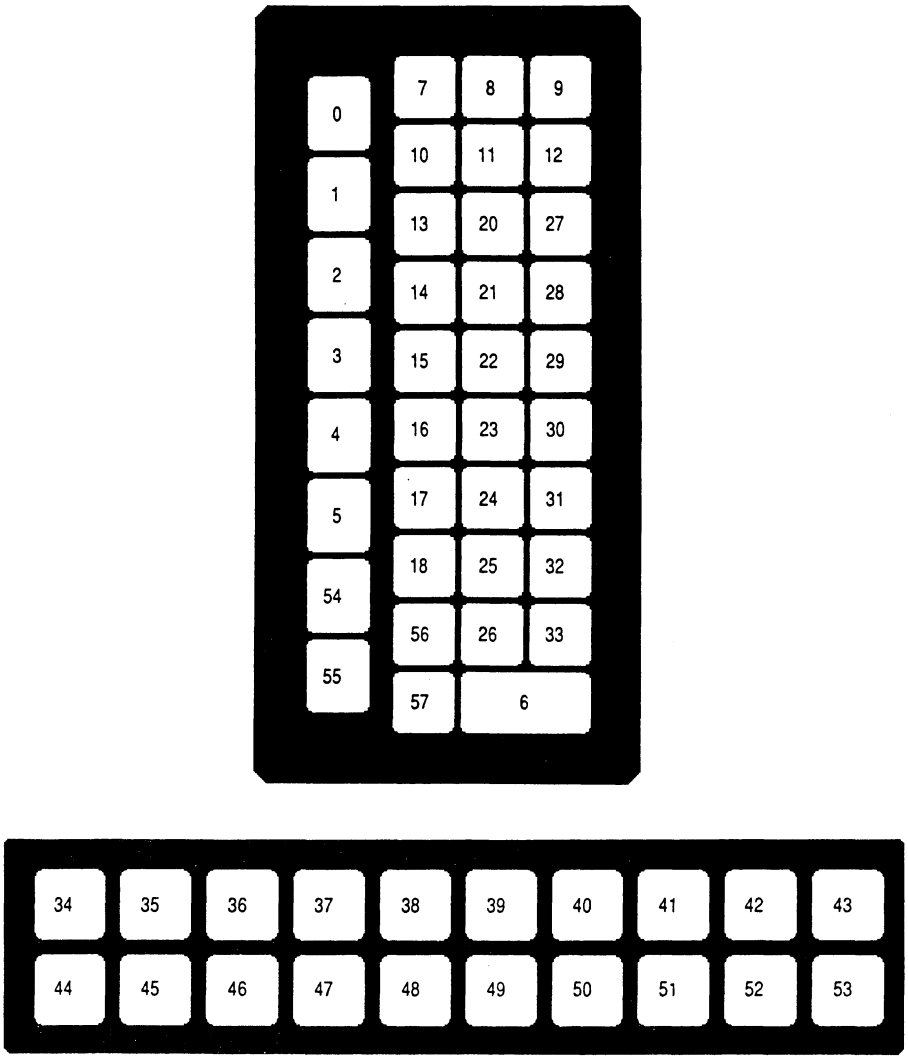


Figure 6-4. 2050/2060 Programmable Keypad and Function Keys

6.4.49 Change to Normal Mode (Touch Screen Only)

Function: Sets the workstation to operate in Normal Mode.

> **NOTE**
> On power-up, the 2000 with touch screen is always in Normal Mode.

Hazeltine emulation: 7E 3A 0
ANSI emulation: ESC [41; 0 p

6.4.50 Change to Touch Screen Mode (Touch Screen Only)

Function: Sets the workstation to operate in Touch Screen Mode. In Touch Screen Mode, codes returned are preceded by "ESC T."

Hazeltine emulation: 7E 3A 1
ANSI emulation: ESC [41; 1 p

6.4.51 Set Programmable Touch Screen Zone (Touch Screen Only)

Function: Configures a touch screen zone to transmit a stored screen when pressed.

Hazeltine emulation: 7E 39 <zone> <screen>
ANSI emulation: ESC [40;<zone>;<screen> p

where: <zone> is the touch screen zone to be configured
<screen> is the screen to send (1 - 255)

6.4.52 **Define Zone or Zones (Touch Screen Only)**

Function: Sets a zone or contiguous area of zones to return a common user-specified code.

Hazeltine emulation: 7E 3B <upper left zone> <lower right zone> <code>
ANSI emulation: ESC [42; <upper left zone>;<lower right zone>;<code> p

where: <upper left zone> is the upper left boundary of the zone to be defined
<lower right zone> is the lower right boundary of the zone to be defined
<code> is a decimal (1-255) or hexadecimal (1-FF) number

Example 1: ESC [42; 35;47;12 p

This example causes contiguous touch screen zones 35-37 and 45-47 to return the value 12 when pressed. Other zones remain unaffected.

Example 2: ESC [42; 71;71;BB p

This example causes the single touch screen zone 71 to return the value BB when pressed. Other zones remain unaffected.

6.5 Sample Screen Display

This example illustrates how to create a simple screen display by transmitting a sequence of characters to the workstation. This display prints the letters "WARNING" in quad size, reverse video inside a box.

ANSI

```
<ESC>[2;1;6;19;12;55p      -- Draw a box<CR>
<ESC>[7;20H                -- Position cursor inside of box<CR>
<ESC>[1;52;7m              -- Select quad size, reverse video<CR>
WARNING
```

Hazeltine 1500

To create the same screen as the ANSI example, send the following characters to the workstation. Note that the hexadecimal representations of the ASCII characters are listed, not the ASCII characters themselves.

```
7E 09 01 12 05 36 0B       -- Draw a box<CR>
7E 11 13 06                -- Position cursor inside of box<CR>
7E 34                      -- Select quad size
7E 07                      -- Reverse video <CR>
WARNING                    -- Message on screen
```

7.1 INTRODUCTION

The 2000 Workstation communications capability allows data to be transferred between the workstation and a host device. The workstation is equipped with two RS-232C or RS-485 serial ports and a parallel port configurable for input or output. Pin numbers and signals for these ports are given in Section 2.12.

7.2 COMMUNICATIONS FORMAT

The communications ports available on the workstation support asynchronous serial data transfer using ASCII codes. Data is transmitted and received at the same baud rate, and this parameter can be set to 300, 600, 1200, 2400, 4800, 9600, or 19200 for each available port.

Each transmitted character includes one start bit, seven or eight data bits, one or no parity bit, and one stop bit (see Figure 7-1). The number of data bits and the parity are selected in the configuration menus (see Section 3.4).

Start Bit	Data Bit 0	Data Bit 1	Data Bit 2	Data Bit 3	Data Bit 4	Data Bit 5	Data Bit 6	Parity Bit	Stop Bit

Seven Data Bits per Character

Start Bit	Data Bit 0	Data Bit 1	Data Bit 2	Data Bit 3	Data Bit 4	Data Bit 5	Data Bit 6	Data Bit 7	Stop Bit

Eight Data Bits per Character - Parity Disabled

Start Bit	Data Bit 0	Data Bit 1	Data Bit 2	Data Bit 3	Data Bit 4	Data Bit 5	Data Bit 6	Data Bit 7	Parity Bit	Stop Bit

Eight Data Bits per Character - Parity Enabled

Figure 7-1. Character Format

Seven data bits per character with parity disabled is not allowed.

The parity condition can be set to one, zero, even, or odd, and the workstation can operate in full or half-duplex modes.

7.3 PARITY CHECKING

The configuration menus allow the user to select whether parity will be employed or not. If parity is employed, the user can select the parity condition to be used: always one, always zero, even, or odd.

When the workstation transmits any character (i.e., when a key is pressed), the settings in the configuration menus will determine the character format and the value of the parity bit (if any).

The workstation only checks parity on received data if odd or even parity is selected. If a character is received with an incorrect parity bit, a parity error symbol (P_E) will be shown on the video display at the cursor position and an audible alarm (beep) will sound.

7.4 FULL AND HALF-DUPLEX OPERATION

When operating in full-duplex terminal mode, the workstation will only display information and execute commands that are received from the host device. Alternately, information and commands can be entered using the workstation's keyboard or optional keypad, and echoed back to the workstation from the host device. In full-duplex mode, the RTS signal will not go high until a key is pressed on the workstation, unless RTS/CTS handshaking is enabled.

When operating in half-duplex terminal mode, the workstation will display information and execute commands that originate from the host device.

7.5 HALF-DUPLEX OPERATION WITH A MODEM

Modem control signals are used if the DCD input to the workstation is high (active). This indicates that the workstation is connected to a modem.

When data is entered the workstation's RTS signal to the modem is set high.

If DCD is detected as high (active), the workstation waits for CTS to go high (active) before transmitting the character. If DCD is low (inactive), the character is transmitted immediately. The workstation holds its RTS signal high (active) and entered data is transmitted until one of the following characters is entered:

CR (0DH)
ETX (03H)
EOT (04H)

After one of the above characters is transmitted, the workstation's RTS signal is made low (inactive) and the modem enters the receive mode. The sequence is repeated every time data is entered.

7.6 INPUT BUFFER OVERFLOW PROTECTION

When the workstation receives a character, it is stored in an input buffer until processed. In unusual circumstances, if the workstation receives characters faster than it can process them, the input buffer can fill. If the workstation's input buffer becomes full and more characters are received, those additional characters will be lost because there is no room to store them.

To prevent this, operate the workstation/host communications link at a baud rate low enough to give the workstation plenty of time to process a character before another is received.

Another way to prevent input buffer overflow is to send the workstation fill characters between valid data. The <NUL> character (00H) is used as the fill character. When received by the workstation, the <NUL> character is ignored. Commands for operations which require a relatively long time for the workstation to perform should be followed by fill characters if this method is used.

Table 7-1. Commands Whose Use May Require Input Buffer Protection

Clear Screen	Draw Box
Clear Foreground	Draw Vertical Line
Clear to End of Line	Draw Horizontal Line
Clear to End of Screen	Execute Screen
Clear to End of Screen (background spaces)	Clear Line
Delete Line	Draw Bar (Up, Down, Left, Right)
Insert Line	
Display of double and quad-size characters	Insert Spaces
Clear to Beginning of Line	Delete Characters
Clear to beginning of Screen	Pause

A preferred method for preventing input buffer overflow is to use either RTS/CTS or XON/XOFF control characters when operating in full-duplex mode. If XON/XOFF generation is enabled, and if there are fewer than 32 free bytes remaining in the input buffer, the XOFF control character will be sent to the host device at this time. When the XOFF signal is received, the host device should stop transmitting. When the buffer contains more than 1000 free bytes, the XON control character will be sent to the host device. Transmission can then resume. The following characters are used as the XON/XOFF characters:

XON = DC1 (11H) XOFF = DC3 (13H)

> **NOTE**
> XON/XOFF should not be used in Hazeltine mode.

If RTS/CTS handshaking is selected from the configuration menus (see Section 3.4), the workstation must have an active CTS before it will transmit data, and will activate RTS when it is able to receive data.

The 2000 Series Industrial Workstations may be rack or panel mounted. Because the workstation is designed to be secured with mounting extrusions, no holes need to be drilled in the panel or rack mount adapter plate. Instructions for mounting the workstation and an optional sealed keyboard are given below.

A.1 PANEL-MOUNTING THE 2000 INDUSTRIAL WORKSTATION AND OPTIONAL SEALED KEYBOARD

1. Cut a hole 11.97 inches wide by 9.97 inches high into the panel to prepare it for the workstation.

2. Insert the workstation into the hole from the front of the panel.

3. Insert the sealed keyboard into the hole from the front of the panel.

4. Hold the workstation flush to the panel and secure it with the mounting extrusions, as shown in Figure A-1 on the following page.

A.2 RACK-MOUNTING THE 2000 INDUSTRIAL WORKSTATION AND OPTIONAL SEALED KEYBOARD

The 2000-RMA rack mount adapter plate and 2000-KBA keyboard adapter plate are required for mounting the workstation and optional sealed keyboard in an equipment rack. The adapter plate is pre-cut; simply mount the workstation and keyboard to the plates according to the panel-mount instructions given above.

After mounting the workstation and keyboard to the adapter plates, secure each unit to a standard equipment rack with eight 10-32 studs.

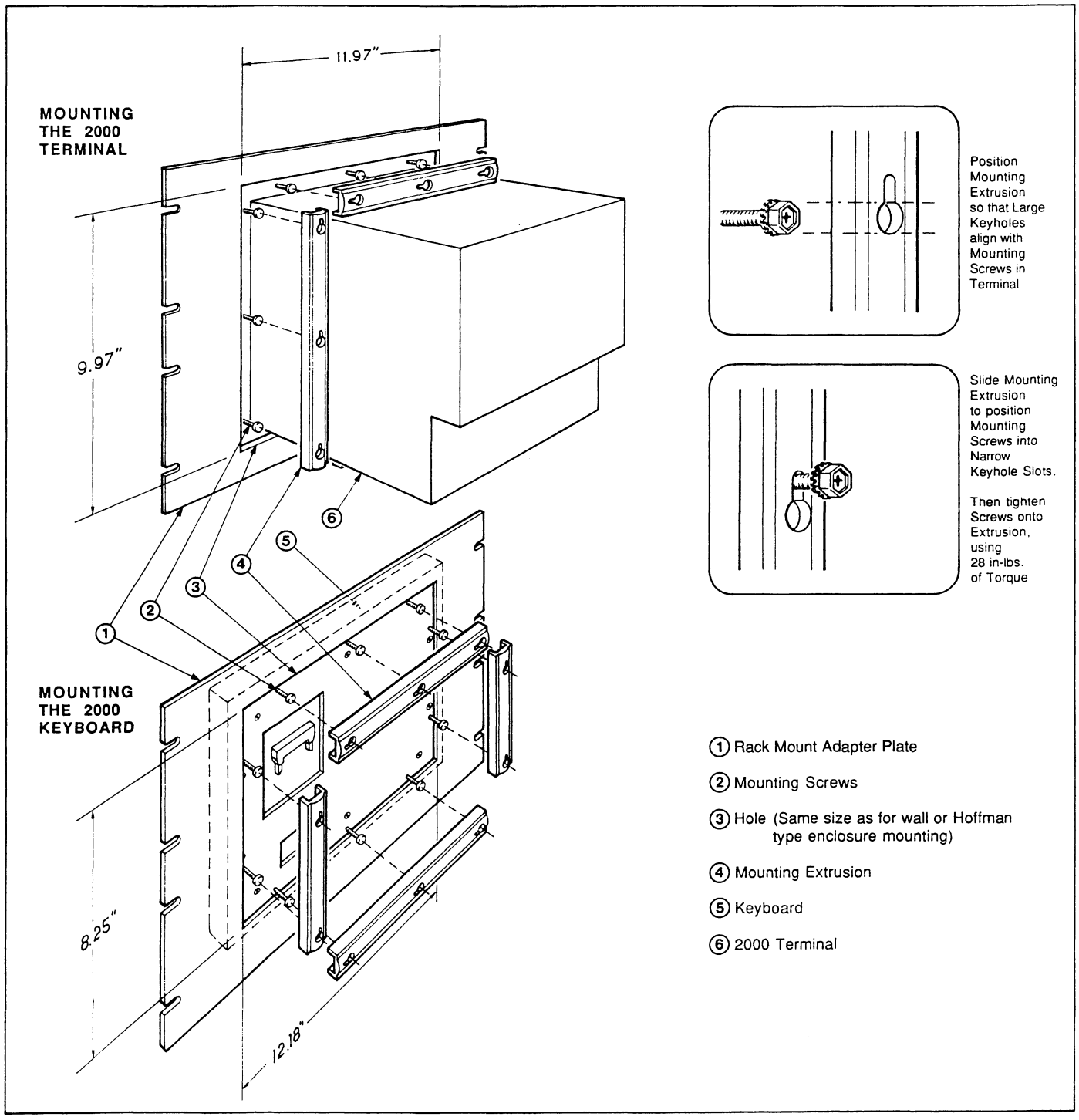
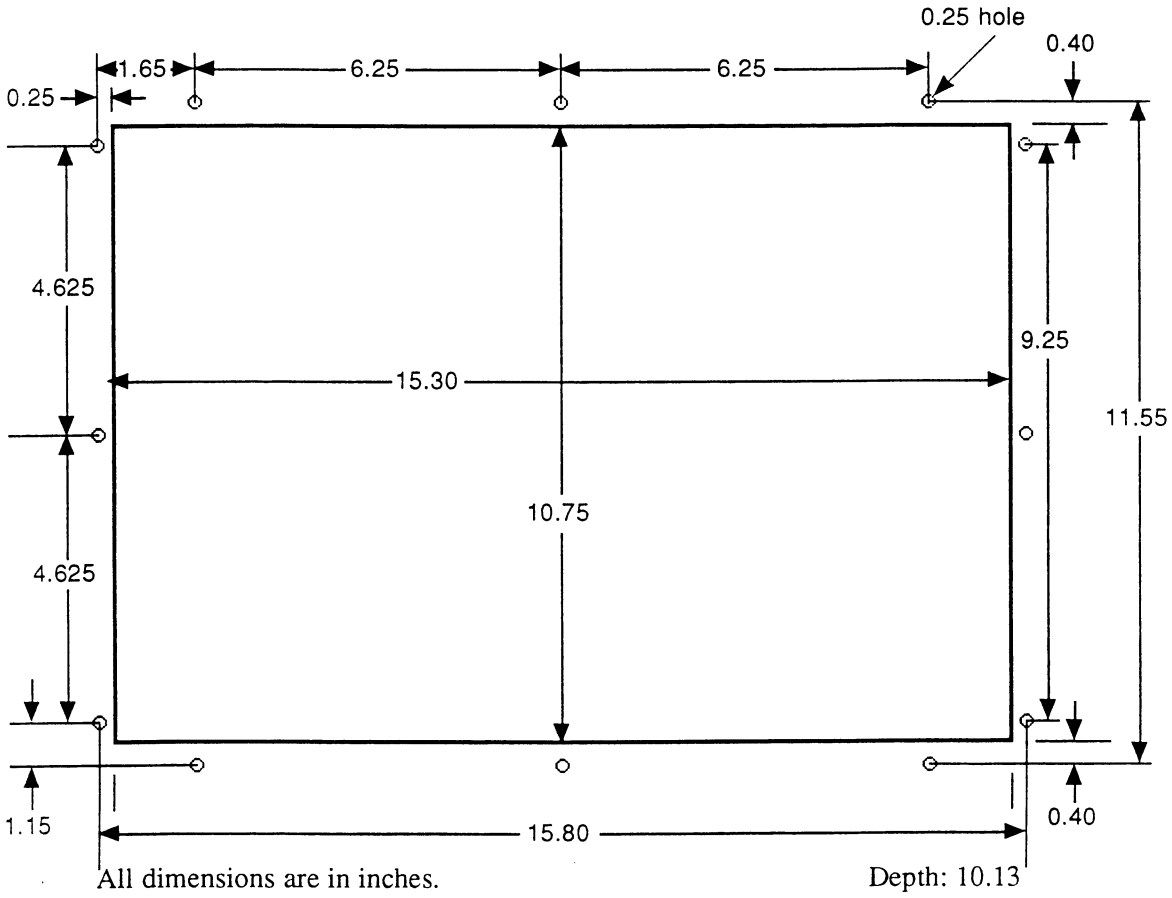


Figure A-1. Mounting the 2000 Industrial Workstation and Optional Sealed Keyboard

A.3 MOUNTING THE 2005

The cutout dimensions needed to mount the 2005 Workstations are shown in Figure A-2 below.

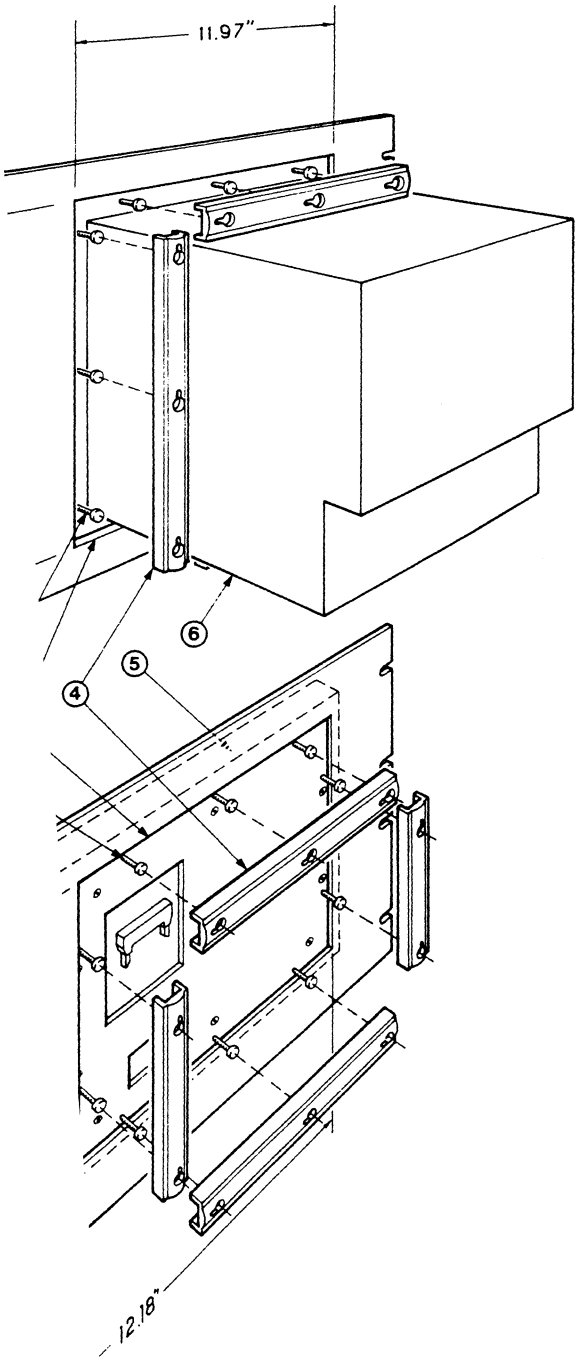


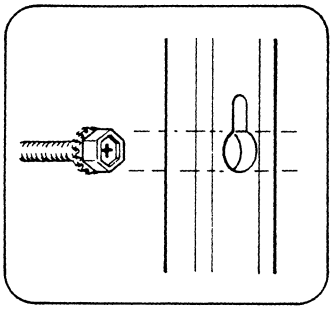
All dimensions are in inches. Depth: 10.13

Torque 10-32 mounting nuts to 25 inch-lbs.

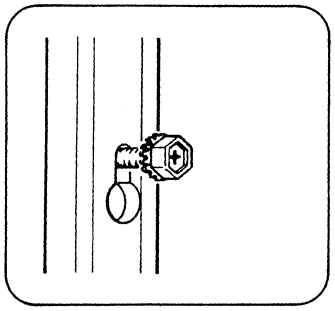
Depth listed does not account for any cabling that may extend beyond the unit.

Figure A-2. 2005 Mounting Dimensions





Position Mounting Extrusion so that Large Keyholes align with Mounting Screws in Terminal

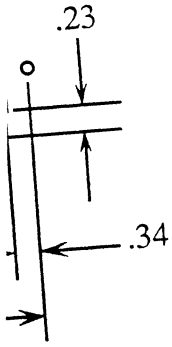


Slide Mounting Extrusion to position Mounting Screws into Narrow Keyhole Slots.

Then tighten Screws onto Extrusion, using 28 in-lbs. of Torque

1. Rack Mount Adapter Plate
2. Mounting Screws
3. Hole (Same size as for wall or Hoffman type enclosure mounting)
4. Mounting Extrusion
5. Keyboard
6. 2000 Terminal

own in



nes.

nit.

ıting the 2000 Industrial Workstation and Optional Sealed Keyboard

A character or a number representing a character (character code) is sent to the display. The symbol displayed depends on the character set selected. The characters and the corresponding scharacter codes are shown along the top axis of the table (see Table B-2). The character set choices are shown along the left axis of the table. The designation of the character set differs depending on whether or not a Xycom firmware chip containing Operator Interface Language (OIL) is installed in the terminal.

If OIL firmware is not installed, the base terminal character set is selected. The base terminal character set is indicated along the left axis using attribute byte 2, bits 2-0, and attribute byte 1, bit 1 (see Table B-1 on the following page).

Table B-1.

Attribute Bytes 2 and 1 Base Terminal Character Sets

	Attribute Byte 2 Bits 2 - 0	Byte 1 Bit 1	Set/Reset Attribute Code
Regular	000	0	50
Double-Wide	000	1	54
Double-High	001	0	51
Double-Size	001	1	55
Quad-Size	010	0	52
Process Graphics	011	0	53
Utility Graphics	111	0	57

Table B-2. Process Graphics Chart

CHARACTER		NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEX		00	01	02	03	04	05	06	07	08
DECIMAL		0	1	2	3	4	5	6	7	8
CHARACTER SET										
BASE* TERM.	OIL OPTION									
000	REG.	N_U (1)	S_H (1)	S_X (1)	E_X (1)	E_T (1)	E_Q (1)	A_K (1)	B_L (1)	B_S (1)
000**	DW	N_U (2)	S_H (2)	S_X (2)	E_X (2)	E_T (2)	E_Q (2)	A_K (2)	B_L (2)	B_S (2)
001	DH									
001**	DS									
010	QS									
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES:

1 - ONLY DISPLAYABLE IN BASE TERMINAL WHEN CONFIGURATION OPTION "DISPLAY CONTROL CODES" IS ENABLED; ALWAYS DISPLAYED WITH OIL OPTION

2 - NOT POSSIBLE ON BASE TERMINAL

* - ATTRIBUTE BYTE 2, BITS 2-0

** - ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

Table B-2. Process Graphics Chart - Continued

CHARACTER		NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEX		09	0A	0B	0C	0D	0E	0F	10	11
DECIMAL		9	10	11	12	13	14	15	16	17
CHARACTER SET										
BASE* TERM.	OIL OPTION									
000	REG.	H_T (1)	L_F (1)	V_T (1)	F_F (1)	C_R (1)	S_O (1)	S_I (1)	D_L (1)	D_1 (1)
000**	DW	H_T (2)	L_F (2)	V_T (2)	F_F (2)	C_R (2)	S_O (2)	S_I (2)	D_L (2)	D_1 (2)
001	DH									
001**	DS									
010	QS									
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES:

1 - ONLY DISPLAYABLE IN BASE TERMINAL WHEN CONFIGURATION OPTION "DISPLAY CONTROL CODES" IS ENABLED; ALWAYS DISPLAYED WITH OIL OPTION EXCEPT FOR "CR," WHICH CAUSES A CARRIAGE RETURN

2 - NOT POSSIBLE ON BASE TERMINAL

* - ATTRIBUTE BYTE 2, BITS 2-0

** - ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

Table B-2. Process Graphics Chart - Continued

CHARACTER		NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEX		12	13	14	15	16	17	18	19	1A
DECIMAL		18	19	20	21	22	23	24	25	26
CHARACTER SET										
BASE* TERM.	OIL OPTION									
000	REG.	D_2 (1)	D_3 (1)	D_4 (1)	N_K (1)	S_Y (1)	E_B (1)	C_N (1)	E_M (1)	S_B (1)
000**	DW	D_2	D_3	D_4	N_K	S_Y	E_B	C_N	E_M	S_B
001	DH									
001**	DS									
010	QS									
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES:

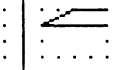
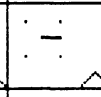
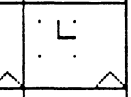
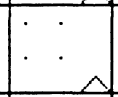
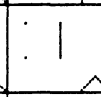
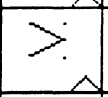
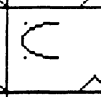
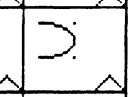
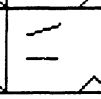
1 - ONLY DISPLAYABLE IN BASE TERMINAL WHEN CONFIGURATION OPTION "DISPLAY CONTROL CODES" IS ENABLED; ALWAYS DISPLAYED WITH OIL OPTION

2 - NOT POSSIBLE ON BASE TERMINAL

* - ATTRIBUTE BYTE 2, BITS 2-0

** - ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

Table B-2. Process Graphics Chart - Continued

CHARACTER		NONE	NONE	NONE	NONE	NONE	SPACE	!	"	#
HEX		1B	1C	1D	1E	1F	20	21	22	23
DECIMAL		27	28	29	30	31	32	33	34	35
CHARACTER SET										
BASE* TERM.	OIL OPTION									
000	REG.	E_C (1)	F_S (1)	G_S (1)	R_S (1)	U_S (1)	SPACE	!	"	#
000**	DW	E_C (2)	F_S (2)	G_S (2)	R_S (2)	U_S (2)	SPACE	!	"	#
001	DH						SPACE	!	"	#
001**	DS						SPACE	!	"	#
010	QS						SPACE	!	"	#
011	G1						SPACE			
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES:

1 - ONLY DISPLAYABLE IN BASE TERMINAL WHEN CONFIGURATION OPTION "DISPLAY CONTROL CODES" IS ENABLED; ALWAYS DISPLAYED WITH OIL OPTION

2 - NOT POSSIBLE ON BASE TERMINAL

* - ATTRIBUTE BYTE 2, BITS 2-0

** - ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ - CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

Table B-2. Process Graphics Chart - Continued

CHARACTER		$	%	&	'	(	)	*	+	,
HEX		24	25	26	27	28	29	2A	2B	2C
DECIMAL		36	37	38	39	40	41	42	43	44
CHARACTER SET										
BASE* TERM.	OIL OPTION									
000	REG.	$	%	&	'	(	)	*	+	,
000**	DW	$	%	&	'	(	)	*	+	,
001	DH	$	%	&	'	(	)	*	+	,
001**	DS	$	%	&	'	(	)	*	+	,
010	QS	$	%	&	'	(	)	*	+	,
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

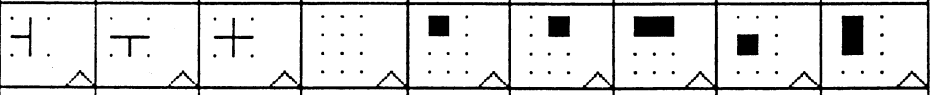
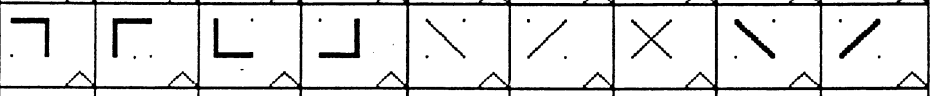
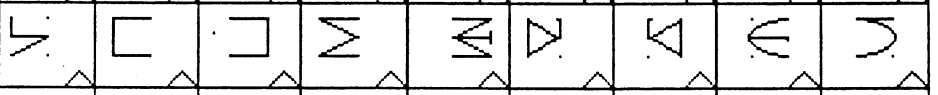
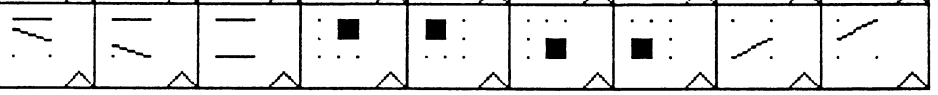
NOTES:

* - ATTRIBUTE BYTE 2, BITS 2-0

** - ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ - CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

Table B-2. Process Graphics Chart - Continued

CHARACTER		-	.	/	0	1	2	3	4	5
HEX		2D	2E	2F	30	31	32	33	34	35
DECIMAL		45	46	47	48	49	50	51	52	53
CHARACTER SET										
BASE* TERM.	OIL OPTION									
000	REG.	-	.	/	0	1	2	3	4	5
000**	DW	—	·	⁄	Ø	1	2	3	4	5
001	DH	-	·	/	Ø	1	2	3	4	5
001**	DS	—	·	/	Ø	1	2	3	4	5
010	QS	—	·	⁄	Ø	1	2	3	4	5
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES:

* - ATTRIBUTE BYTE 2, BITS 2-0

** - ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ - CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

Table B-2. Process Graphics Chart - Continued

CHARACTER		6	7	8	9	:	;	<	=	>
HEX		36	37	38	39	3A	3B	3C	3D	3E
DECIMAL		54	55	56	57	58	59	60	61	62
CHARACTER SET										
BASE* TERM.	OIL OPTION									
000	REG.	6	7	8	9	:	;	<	=	>
000**	DW	6	7	8	9	:	;	<	=	>
001	DH	6	7	8	9	:	;	<	=	>
001**	DS	6	7	8	9	:	;	<	=	>
010	QS	6	7	8	9	:	;	<	=	>
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES:

* - ATTRIBUTE BYTE 2, BITS 2-0

** - ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ - CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

Table B-2. Process Graphics Chart - Continued

CHARACTER		?	@	A	B	C	D	E	F	G
HEX		3F	40	41	42	43	44	45	46	47
DECIMAL		63	64	65	66	67	68	69	70	71
CHARACTER SET										
BASE* TERM.	OIL OPTION									
000	REG.	?	@	A	B	C	D	E	F	G
000**	DW	?	@	A	B	C	D	E	F	G
001	DH	?	@	A	B	C	D	E	F	G
001**	DS	?	@	A	B	C	D	E	F	G
010	QS	?	@	A	B	C	D	E	F	G
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES:

* - ATTRIBUTE BYTE 2, BITS 2-0

** - ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ - CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

Table B-2. Process Graphics Chart - Continued

CHARACTER		H	I	J	K	L	M	N	O	P
HEX		48	49	4A	4B	4C	4D	4E	4F	50
DECIMAL		72	73	74	75	76	77	78	79	80
CHARACTER SET										
BASE* TERM.	OIL OPTION									
000	REG.	H	I	J	K	L	M	N	O	P
000**	DW	H	I	J	K	L	M	N	O	P
001	DH	H	I	J	K	L	M	N	O	P
001**	DS	H	I	J	K	L	M	N	O	P
010	QS	H	I	J	K	L	M	N	O	P
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES:

* - ATTRIBUTE BYTE 2, BITS 2-0

** - ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ - CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

Table B-2. Process Graphics Chart - Continued

CHARACTER		Q	R	S	T	U	V	W	X	Y
HEX		51	52	53	54	55	56	57	58	59
DECIMAL		81	82	83	84	85	86	87	88	89
CHARACTER SET										
BASE* TERM.	OIL OPTION									
000	REG.	Q	R	S	T	U	V	W	X	Y
000**	DW	Q	R	S	T	U	V	W	X	Y
001	DH	Q	R	S	T	U	V	W	X	Y
001**	DS	Q	R	S	T	U	V	W	X	Y
010	QS	Q	R	S	T	U	V	W	X	Y
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES:

* - ATTRIBUTE BYTE 2, BITS 2-0

** - ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ - CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

Table B-2. Process Graphics Chart - Continued

CHARACTER		Z	[	\	]		_	`	a	b
HEX		5A	5B	5C	5D	5E	5F	60	61	62
DECIMAL		90	91	92	93	94	95	96	97	98
CHARACTER SET										
BASE* TERM.	OIL OPTION									
000	REG.									
000**	DW									
001	DH									
001**	DS									
010	QS									
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

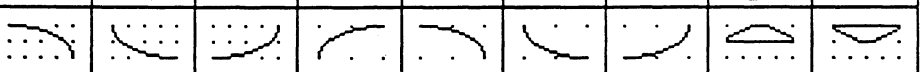
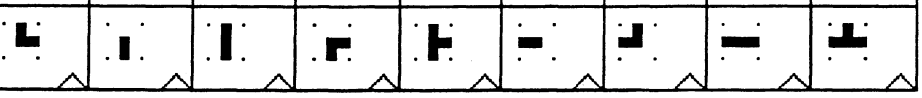
NOTES:

* - ATTRIBUTE BYTE 2, BITS 2-0

** - ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ - CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

Table B-2. Process Graphics Chart - Continued

CHARACTER		c	d	e	f	g	h	i	j	k
HEX		63	64	65	66	67	68	69	6A	6B
DECIMAL		99	100	101	102	103	104	105	106	107
CHARACTER SET										
BASE* TERM.	OIL OPTION									
000	REG.	c	d	e	f	g	h	i	j	k
000**	DW	c	d	e	f	g	h	i	j	k
001	DH	c	d	e	f	g	h	i	j	k
001**	DS	c	d	e	f	g	h	i	j	k
010	QS	c	d	e	f	g	h	i	j	k
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES:

* - ATTRIBUTE BYTE 2, BITS 2-0

** - ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ - CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

Table B-2. Process Graphics Chart - Continued

CHARACTER		l	m	n	o	p	q	r	s	t
HEX		6C	6D	6E	6F	70	71	72	73	74
DECIMAL		108	109	110	111	112	113	114	115	116
CHARACTER SET										
BASE* TERM.	OIL OPTION									
000	REG.	l	m	n	o	p	q	r	s	t
000**	DW	l	m	n	o	p	q	r	s	t
001	DH	l	m	n	o	p	q	r	s	t
001**	DS	l	m	n	o	p	q	r	s	t
010	QS	l	m	n	o	p	q	r	s	t
011	G1									
N/A	G2						(1)	(2)	(3)	(4)
N/A	G3									
N/A	G4									
111	N/A									

NOTES:

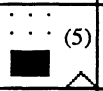
* - ATTRIBUTE BYTE 2, BITS 2-0

** - ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ - CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

(x) - NUMBER IN PARENTHESES INDICATES HOW HIGH OR WIDE CHARACTER IS RELATIVE TO THE CELL

Table B-2. Process Graphics Chart - Continued

CHARACTER		u	v	w	x	y	z	{		}
HEXIDECIMAL		75	76	77	78	79	7A	7B	7C	7D
DECIMAL		117	118	119	120	121	122	123	124	125
CHARACTER SET										
BASE* TERM.	OIL OPTION									
000	REG.	u	v	w	x	y	z	{	|	}
000**	DW	u	v	w	x	y	z	{	|	}
001	DH	u	v	w	x	y	z	{	|	}
001**	DS	u	v	w	x	y	z	{	|	}
010	QS	u	v	w	x	y	z	{	|	}
011	G1									
N/A	G2	 (5)	 (6)	 (7)	 (8)	 (9)	 (10)	 (11)	 (12)	
N/A	G3									
N/A	G4									
111	N/A									

NOTES:

* - ATTRIBUTE BYTE 2, BITS 2-0

** - ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ - CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

(x) - NUMBER IN PARENTHESES INDICATES HOW HIGH OR WIDE CHARACTER IS RELATIVE TO THE CELL

Table B-2. Process Graphics Chart - Continued

CHARACTER		~	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEXIDECIMAL		7E	7F	80	81	82	83	84	85	86
DECIMAL		126	127	128	129	130	131	132	133	134
CHARACTER SET										
BASE* TERM.	OIL OPTION									
000	REG.	~								
000**	DW	~								
001	DH	~								
001**	DS	~								
010	QS	~								
011	G1									
N/A	G2				(1)	(2)	(3)	(4)	(5)	(6)
N/A	G3									
N/A	G4									
111	N/A									

NOTES:

* - ATTRIBUTE BYTE 2, BITS 2-0

** - ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ - CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

(x) - NUMBER IN PARENTHESES INDICATES HOW HIGH OR WIDE CHARACTER IS RELATIVE TO THE CELL

Table B-2. Process Graphics Chart - Continued

CHARACTER		NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEXIDECIMAL		87	88	89	8A	8B	8C	8D	8E	8F
DECIMAL		135	136	137	138	139	140	141	142	143
CHARACTER SET										
BASE* TERM.	OIL OPTION									
000	REG.									
000**	DW									
001	DH									
001**	DS									
010	QS									
011	G1									
N/A	G2	(7)	(8)	(9)	(10)	(11)	(12)			
N/A	G3									
N/A	G4									
111	N/A									

NOTES:

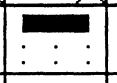
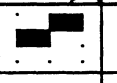
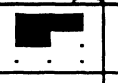
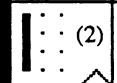
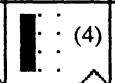
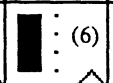
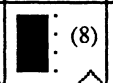
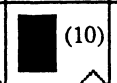
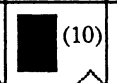
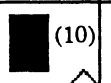
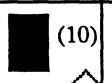
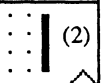
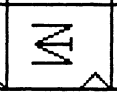
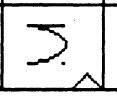
* - ATTRIBUTE BYTE 2, BITS 2-0

** - ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ - CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

(x) - NUMBER IN PARENTHESES INDICATES HOW HIGH OR WIDE CHARACTER IS RELATIVE TO THE CELL

Table B-2. Process Graphics Chart - Continued

CHARACTER		NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEXIDECIMAL		90	91	92	93	94	95	96	97	98
DECIMAL		144	145	146	147	148	149	150	151	152
CHARACTER SET										
BASE* TERM.	OIL OPTION									
000	REG.									
000**	DW									
001	DH									
001**	DS									
010	QS									
011	G1									
N/A	G2	 (2)	 (4)	 (6)	 (8)	 (10)	 (10)	 (10)	 (10)	 (2)
N/A	G3									
N/A	G4									
111	N/A									

NOTES:

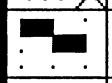
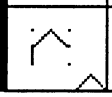
* - ATTRIBUTE BYTE 2, BITS 2-0

** - ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ - CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

(x) - NUMBER IN PARENTHESES INDICATES HOW HIGH OR WIDE CHARACTER IS RELATIVE TO THE CELL

Table B-2. Process Graphics Chart - Continued

CHARACTER		NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEXIDECIMAL		99	9A	9B	9C	9D	9E	9F	A0	A1
DECIMAL		153	154	155	156	157	158	159	160	161
CHARACTER SET										
BASE* TERM.	OIL OPTION									
000	REG.									
000**	DW									
001	DH									
001**	DS									
010	QS									
011	G1									
N/A	G2	 (4)	(6)	(8)	(10)	(10)	(10)	(10)		
N/A	G3									
N/A	G4									
111	N/A									

NOTES:

* - ATTRIBUTE BYTE 2, BITS 2-0

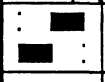
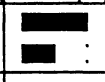
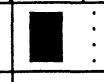
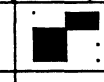
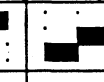
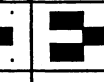
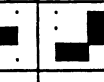
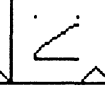
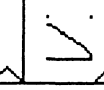
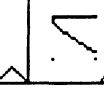
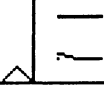
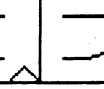
** - ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ - CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

(x) -NUMBER IN PARENTHESES INDICATES HOW HIGH OR WIDE CHARACTER IS RELATIVE TO THE CELL

○ - DOUBLE-WIDE CHARACTERS ARE NOT AVAILABLE FOR CHARACTERS 158-207 ON COLOR UNIT

Table B-2. Process Graphics Chart - Continued

CHARACTER		NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEXIDECIMAL		A2	A3	A4	A5	A6	A7	A8	A9	AA
DECIMAL		162	163	164	165	166	167	168	169	170
CHARACTER SET										
BASE* TERM.	OIL OPTION									
000	REG.									
000**	DW									
001	DH									
001**	DS									
010	QS									
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

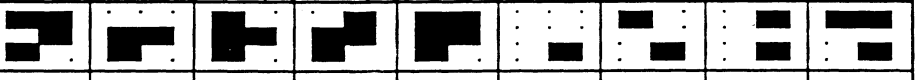
NOTES:

* - ATTRIBUTE BYTE 2, BITS 2-0

** - ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ - CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

Table B-2. Process Graphics Chart - Continued

CHARACTER		NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEXIDECIMAL		AB	AC	AD	AE	AF	BO	B1	B2	B3
DECIMAL		171	172	173	174	175	176	177	178	179
CHARACTER SET										
BASE* TERM.	OIL OPTION									
000	REG.									
000**	DW									
001	DH									
001**	DS									
010	QS									
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

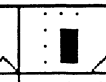
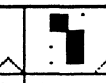
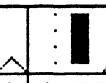
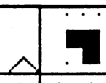
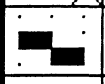
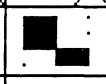
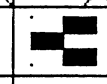
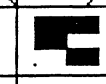
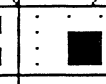
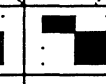
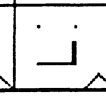
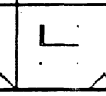
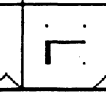
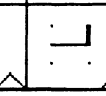
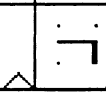
NOTES:

* - ATTRIBUTE BYTE 2, BITS 2-0

** - ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ - CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

Table B-2. Process Graphics Chart - Continued

CHARACTER		NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEXIDECIMAL		B4	B5	B6	B7	B8	B9	BA	BB	BC
DECIMAL		180	181	182	183	184	185	186	187	188
CHARACTER SET										
BASE* TERM.	OIL OPTION									
000	REG.									
000**	DW									
001	DH									
001**	DS									
010	QS									
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

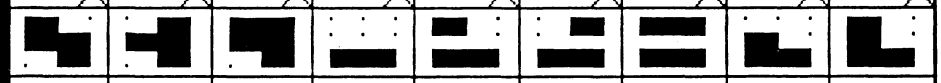
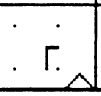
NOTES:

* - ATTRIBUTE BYTE 2, BITS 2-0

** - ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ - CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

Table B-2. Process Graphics Chart - Continued

CHARACTER		NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEXIDECIMAL		BD	BE	BF	CO	C1	C2	C3	C4	C5
DECIMAL		189	190	191	192	193	194	195	196	197
CHARACTER SET										
BASE* TERM.	OIL OPTION									
000	REG.									
000**	DW									
001	DH									
001**	DS									
010	QS									
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES:

* - ATTRIBUTE BYTE 2, BITS 2-0

** - ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ - CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

Table B-2. Process Graphics Chart - Continued

CHARACTER		NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEXIDECIMAL		C6	C7	C8	C9	CA	CB	CC	CD	CE
DECIMAL		198	199	200	201	202	203	204	205	206
CHARACTER SET										
BASE* TERM.	OIL OPTION									
000	REG.									
000**	DW									
001	DH									
001**	DS									
010	QS									
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

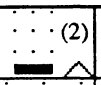
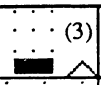
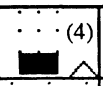
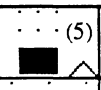
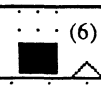
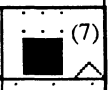
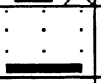
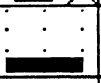
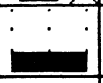
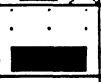
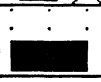
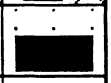
NOTES:

* - ATTRIBUTE BYTE 2, BITS 2-0

** - ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ - CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

Table B-2. Process Graphics Chart - Continued

CHARACTER		NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEXIDECIMAL		CF	D0	D1	D2	D3	D4	D5	D6	D7
DECIMAL		207	208	209	210	211	212	213	214	215
CHARACTER SET										
BASE* TERM.	OIL OPTION									
000	REG.			(1)	 (2)	 (3)	 (4)	 (5)	 (6)	 (7)
000**	DW									
001	DH									
001**	DS									
010	QS									
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES:

* - ATTRIBUTE BYTE 2, BITS 2-0

** - ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ - CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

(x) - NUMBER IN PARENTHESES INDICATES HOW HIGH OR WIDE CHARACTER IS RELATIVE TO THE CELL; CORRESPONDING DOUBLE-WIDE CHARACTERS ARE THE SAME RELATIVE SIZE

Table B-2. Process Graphics Chart - Continued

CHARACTER		NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEXIDECIMAL		D8	D9	DA	DB	DC	DD	DE	DF	E0
DECIMAL		216	217	218	219	220	221	222	223	224
CHARACTER SET										
BASE* TERM.	OIL OPTION									
000	REG.	(8)	(9)	(10)	(11)	(12)				
000**	DW									
001	DH									
001**	DS									
010	QS									
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES:

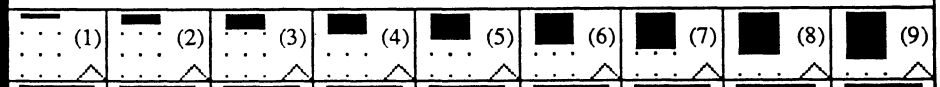
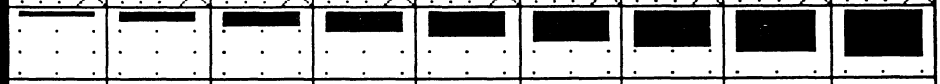
* - ATTRIBUTE BYTE 2, BITS 2-0

** - ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ - CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

(x) - NUMBER IN PARENTHESES INDICATES HOW HIGH OR WIDE CHARACTER IS RELATIVE TO THE CELL; CORRESPONDING DOUBLE-WIDE CHARACTERS ARE THE SAME RELATIVE SIZE

Table B-2. Process Graphics Chart - Continued

CHARACTER		NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEXIDECIMAL		E1	E2	E3	E4	E5	E6	E7	E8	E9
DECIMAL		225	226	227	228	229	230	231	232	233
CHARACTER SET										
BASE* TERM.	OIL OPTION									
000	REG.									
000**	DW									
001	DH									
001**	DS									
010	QS									
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES:

* - ATTRIBUTE BYTE 2, BITS 2-0

** - ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ - CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

(x) - NUMBER IN PARENTHESES INDICATES HOW HIGH OR WIDE CHARACTER IS RELATIVE TO THE CELL; CORRESPONDING DOUBLE-WIDE CHARACTERS ARE THE SAME RELATIVE SIZE

Table B-2. Process Graphics Chart - Continued

CHARACTER		NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEXIDECIMAL		EA	EB	EC	ED	EE	EF	F0	F1	F2
DECIMAL		234	235	236	237	238	239	240	241	242
CHARACTER SET										
BASE* TERM.	OIL OPTION									
000	REG.	(10)	(11)	(12)				(2)	(4)	(6)
000**	DW									
001	DH									
001**	DS									
010	QS									
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES:

* - ATTRIBUTE BYTE 2, BITS 2-0

** - ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ - CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

(x) - NUMBER IN PARENTHESES INDICATES HOW HIGH OR WIDE CHARACTER IS RELATIVE TO THE CELL; CORRESPONDING DOUBLE-WIDE CHARACTERS ARE THE SAME RELATIVE SIZE

Table B-2. Process Graphics Chart - Continued

CHARACTER		NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEXIDECIMAL		F3	F4	F5	F6	F7	F8	F9	FA	FB
DECIMAL		243	244	245	246	247	248	249	250	251
CHARACTER SET										
BASE* TERM.	OIL OPTION									
000	REG.	(8)	(10)	(10)	(10)	(10)	(2)	(4)	(6)	(8)
000**	DW									
001	DH									
001**	DS									
010	QS									
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES:

* - ATTRIBUTE BYTE 2, BITS 2-0

** - ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ - CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

(x) - NUMBER IN PARENTHESES INDICATES HOW HIGH OR WIDE CHARACTER IS RELATIVE TO THE CELL; CORRESPONDING DOUBLE-WIDE CHARACTERS ARE THE SAME RELATIVE SIZE

Table B-2. Process Graphics Chart - Continued

CHARACTER		NONE	NONE	NONE	NONE					
HEXIDECIMAL		FC	FD	FE	FF					
DECIMAL		252	253	254	255					
CHARACTER SET										
BASE* TERM.	OIL OPTION									
000	REG.	(10)	(10)	(10)	(10)					
000**	DW									
001	DH									
001**	DS									
010	QS									
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES:

* - ATTRIBUTE BYTE 2, BITS 2-0

** - ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ - CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

(x) - NUMBER IN PARENTHESES INDICATES HOW HIGH OR WIDE CHARACTER IS RELATIVE TO THE CELL; CORRESPONDING DOUBLE-WIDE CHARACTERS ARE THE SAME RELATIVE SIZE

Table B-2. Process Graphics Chart - Continued

CHARACTER		NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
HEXIDECIMAL		F3	F4	F5	F6	F7	F8	F9	FA	FB
DECIMAL		243	244	245	246	247	248	249	250	251
CHARACTER SET										
BASE* TERM.	OIL OPTION									
000	REG.	(8)	(10)	(10)	(10)	(10)	(2)	(4)	(6)	(8)
000**	DW									
001	DH									
001**	DS									
010	QS									
011	G1									
N/A	G2									
N/A	G3									
N/A	G4									
111	N/A									

NOTES:

* - ATTRIBUTE BYTE 2, BITS 2-0

** - ATTRIBUTE BYTE 1, BIT 1 MUST BE SET FOR THIS MODE

△ - CHARACTER CELL SHOWN LARGER THAN ACTUAL SIZE

(x) - NUMBER IN PARENTHESES INDICATES HOW HIGH OR WIDE CHARACTER IS RELATIVE TO THE CELL; CORRESPONDING DOUBLE-WIDE CHARACTERS ARE THE SAME RELATIVE SIZE

The 2000 Series Industrial Workstations support all VT100 codes except for those listed in this Section. If these codes are received by the terminal, they will be ignored.

Control Characters Not Supported

0E - select G1 character set
0F - select GO character set

Digital Equipment Corporation Private Configuration Commands Not Supported

ESC [? 1 h	- enable application interpretation of cursor keys
ESC [? 2 h	- enable ANSI mode
ESC [? 3 h	- enable 132 column mode
ESC [? 5 h	- enable reverse screen mode
ESC [? 6 h	- enable origin mode
ESC [? 8 h	- enable auto repeat
ESC [? 9 h	- enable interlace
ESC [? 18 h	- print form feed enabled
ESC [? 19 h	- full screen print extent
ESC [? 42 h	- national character set
ESC [? 1 l	- disable application interpretation of cursor keys
ESC [? 2 l	- enable VT52 mode
ESC [? 3 l	- enable 80 column mode
ESC [? 5 l	- enable normal screen
ESC [? 6 l	- enable absolute mode
ESC [? 8 l	- disable auto repeat
ESC [? 9 l	- disable interlace
ESC [? 18 l	- print form feed disable
ESC [? 19 l	- scrolling region print extent
ESC [? 42 l	- multinational character set

Configuration Commands Not Supported

ESC [4 h	- insert mode enable
ESC [12 h	- local echo disabled
ESC [4 l	- replace mode enable
ESC [12 l	- local echo enabled

Select Characters Set Codes Not Supported

ESC (A	- UK G0
ESC (B	- US ASCII G0
ESC (0	- special chars and lines G0
ESC (1	- alternate ROM G0
ESC (2	- alternate ROM and special graphics G0
ESC (A	- UK G1
ESC (B	- US ASCII G1
ESC (0	- special chars and lines G1
ESC (1	- alternate ROM G1
ESC (2	- alternate ROM and special graphics G1
ESC N	- single shift 2
ESC O	- single shift 3

Scrolling Region Command Not Supported

ESC [pt;pb r	- set top and bottom margin

Line Attribute Commands Not Supported

ESC # 3	- double-high top half
ESC # 4	- double-high bottom half
ESC # 5	- single-wide, single-high
ESC # 6	- double-wide, single-high
ESC # 8	- fill screen with e's

Test Commands Not Supported

ESC [1;1 y	- invoke power-up test
ESC [2;2 y	- data loopback test
ESC [2;9 y	- continuous power-up testing
ESC [2;10 y	- continuous loopback test

Keyboard LED Commands Not Supported

ESC [0 q	- all LEDs off
ESC [1 q	- LED 1 on
ESC [2 q	- LED 2 on
ESC [3 q	- LED 3 on
ESC [4 q	- LED 4 on

Aux Keypad Codes in Application Mode Not Generated

-	- ESC 0 m
,	- ESC 0 l

Report Commands Not Supported

ESC [? 15n	- what is printer status
ESC [? 25n	- what is status of user-defined keys
ESC [? 26n	- what is keyboard language

Figure D-1. Default Codes Transmitted by Touch Screen Zones
(Base Terminal)

Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9	Z10
80H	81H	82H	83H	84H	85H	86H	87H	88H	89H
Z11	Z12	Z13	Z14	Z15	Z16	Z17	Z18	Z19	Z20
8AH	8BH	8CH	8DH	8EH	8FH	90H	91H	92H	93H
Z21	Z22	Z23	Z24	Z25	Z26	Z27	Z28	Z29	Z30
94H	95H	96H	97H	98H	99H	9AH	9BH	9CH	9DH
Z31	Z32	Z33	Z34	Z35	Z36	Z37	Z38	Z39	Z40
9EH	9FH	A0H	A1H	A2H	A3H	A4H	A5H	A6H	A7H
Z41	Z42	Z43	Z44	Z45	Z46	Z47	Z48	Z49	Z50
A8H	A9H	AAH	ABH	ACH	ADH	AEH	AFH	B0H	B1H
Z51	Z52	Z53	Z54	Z55	Z56	Z57	Z58	Z59	Z60
B2H	B3H	B4H	B5H	B6H	B7H	B8H	B9H	BAH	BBH
Z61	Z62	Z63	Z64	Z65	Z66	Z67	Z68	Z69	Z70
BCH	BDH	BEH	BFH	C0H	C1H	C2H	C3H	C4H	C5H
Z71	Z72	Z73	Z74	Z75	Z76	Z77	Z78	Z79	Z80
C6H	C7H	C8H	C9H	CAH	CBH	CCH	CDH	CEH	CFH

KEY:

Z = Zone (e.g., Z1 = touch screen zone 1)

Table D-1. Process Graphic Symbols

Hex Value	ASCII Character	Process Control Symbol
20H		4x4 space
21H	!	motor in 4x3 cell
22H	"	not used
23H	#	left tank top in 4x1 cell
24H	$	right tank top in 4x1 cell
25H	%	small diamond in 4x2 cell
26H	&	left tank bottom in 4x1 cell
27H	'	right tank bottom in 4x1 cell
28H	(	left arrow in 4x2 cell
29H	)	right arrow in 4x2 cell
2AH	*	small box in 4x2 cell
2BH	+	up valve in 4x2 cell
2CH	,	right/left facing valve in 4x2 cell
2DH	-	pump/compressor in 4x2 cell
2EH	.	up arrow in 4x2 cell
2FH	/	down arrow in 4x2 cell
30H	0	small circle in 4x2 cell
31H	1	circuit breaker type 1 in 2x4 cell
32H	2	fuse in 2x4 cell
33H	3	disconnect in 3x4 cell
34H	4	pump/blower in 4x2 cell
35H	5	circuit breaker type 2 in 4x2 cell
36H	6	left turbine in 3x2 cell
37H	7	right turbine in 3x2 cell
38H	8	left medium box in 4x2 cell
39H	9	right medium box in 4x2 cell
3AH	;	left medium circle in 4x3 cell
3BH	:	right medium circle in 4x3 cell
3CH	<	mini circle in 2x1 cell
3DH	=	mini left arrow in 2x1 cell
3EH	>	mini right arrow in 2x1 cell
3FH	?	mini up arrow in 2x1 cell
40H	@	mini down arrow in 2x1 cell
41H	A	motor
42H	B	large circle (left)
43H	C	large circle (right)
44H	D	tank top (left)
45H	E	tank top (right)
46H	F	small diamond
47H	G	large diamond (left)

Table D-1. Process Graphic Symbols (cont.)

Hex Value	ASCII Character	Process Control Symbol
48H	H	large diamond (right)
49H	I	tank bottom (left)
4AH	J	tank bottom (right)
4BH	K	left arrow
4CH	L	right arrow
4DH	M	small box
4EH	N	up facing valve
4FH	O	right/left facing valve
50H	P	pump/compressor
51H	Q	up arrow
52H	R	down arrow
53H	S	small circle
54H	T	transformer
55H	U	circuit breaker (type 1)
56H	V	fuse
57H	W	disconnect
58H	X	pump/blower
59H	Y	circuit breaker (type 2)
5AH	Z	turbine (left)
5BH	[	turbine (right)
5CH	\	large box (left)
5DH	]	large box (right)
5EH	^	medium box (left)
5FH	_ (underscore)	medium box (right)
60H	` (grave)	medium circle (left)
61H	a	medium circle (right)
62H	b	top left 1/4 of large circle in 4x2 cell
63H	c	top right 1/4 of large circle in 4x2 cell
64H	d	bottom left 1/4 of large circle in 4x2 cell
65H	e	bottom right 1/4 of large circle in 4x2 cell
66H	f	top left 1/4 of small circle in 2x1 cell
67H	g	top right 1/4 of small circle in 2x1 cell
68H	h	bottom left 1/4 of small circle in 2x1 cell
69H	i	bottom right 1/4 of small circle in 2x1 cell
6AH	j	small tank top in 4x1 cell
6BH	k	small tank bottom in 4x1 cell
6CH	l	mini tank top in 2x1 cell
6DH	m	mini tank bottom in 2x1 cell
6EH	n	mini diamond in 2x1 cell
6FH	o	mini box in 2x1 cell

Table D-1. Process Graphic Symbols (cont.)

Hex Value	ASCII Character	Process Control Symbol
70H	p	mini right valve in 2x1 cell
71H	q	mini up valve in 2x1 cell
72H	r	mini motor in 2x2 cell
73H	s	mini pump/blower in 2x1 cell
74H	t	mini transformer in 2x2 cell
75H	u	mini circuit breaker type 1 in 1x2 cell
76H	v	mini fuse in 1x2 cell
77H	w	mini disconnect in 1x2 cell
78H	x	mini blower/compressor in 2x1 cell
79H	y	mini circuit breaker type 2 in 2x1 cell
7AH	z	mini left turbine in 1x1 cell
7BH	(	mini right turbine in 1x1 cell

Table D-2. Utility Graphics

Utility Graphics	Description
32-79 (20-4F Hex)	Process Graphics Pieces
80-87 (50-57 Hex)	Process Graphic Connectors (Thin)
88-95 (58-5F Hex)	Process Graphic Connectors (Thick)
96-111 (60-6F Hex)	Thick Line Graphics
112-175 (70-AF Hex)	Process Graphic Pieces
176-187 (B0-BB Hex)	Miscellaneous Connectors

Table D-3. Remote Commands
(Hazeltine 1500 Emulation)

REMOTE COMMANDS	ASCII	HEX
Control Characters		
Bell	<BEL>	07
Backspace	<BS>	08
Cursor to Next Foreground Field	<HT>	09
Linefeed	<LF>	0A
Carriage Return	<CR>	0D
Configuration Commands		
Enable Application Mode	~ .	7E 2E
Disable Application Mode	~ /	7E 2F
Cursor Off	~<S0H>	7E 01
Cursor On	~<STX>	7E 02
Scrolling Off	~<BEL>	7E 07
Scrolling On	~<BS>	7E 08
Unlock Keyboard	~<ACK>	7E 06
Lock Keyboard	~<NAK>	7E 15
Enable Printer Port	~ *	7E 2A
Disable Printer Port	~ +	7E 2B
Enable Screen Display	~ (	7E 28
Disable Screen Display	~)	7E 29
Attribute Commands		
Set/Reset Attributes	~6<attribute#>	7E 36 <attribute #>
Change Char. Attributes	~<ETX> <attr-1> <attr-2>	7E 03 <attr-1> <attr-2>
Cursor Movement Commands		
Cursor Right (no scroll)	<DLE>	10
Return Cursor Position	~<ENQ>	7E 05
Cursor Down (no scroll)	~<VT>	7E 0B
Cursor Up	~<FF>	7E 0C
Cursor to X,Y	~<DC1> X Y	7E 11 X Y
Home Cursor	~<DC2>	7E 12

Table D-3. Remote Commands (cont.)
(Hazeltine 1500 Emulation)

REMOTE COMMANDS	ASCII	HEX
Clear Commands		
Clear to EOL with Background Spaces	~<SI>	7E 0F
Clear to EOS with Background Spaces	~<ETB>	7E 17
Clear to EOS with Foreground Spaces	~<CAN>	7E 18
Clear Foreground	~<GS>	7E 1D
Clear Screen	~<FS>	7E 1C
Background Field Follows	~<EM>	7E 19
Foreground Field Follows	~<US>	7E 1F
Delete Commands		
Delete Line	~<DC3>	7E 13
Insert Line	~<SUB>	7E 1A
Draw Commands		
Draw Box	~<HT> <char> <xstart> <ystart> <xend> <yend>	7E 09 <char> <xstart> <ystart> <xend> <yend>
Draw Vertical Line (upward)	~<LF> <char> <xstart> <ystart> <length>	7E 0A <char> <xstart> <ystart> <length>
Draw Horizontal Line (left to right)	~<CR> <char> <xstart> <ystart> <length>	7E 0D <char> <xstart> <ystart> <length>
Draw Bar Chart	~<S0> <xstart> <ystart> <length1> <length2>	7E 0E <xstart> <ystart> <length1> <length2>
Draw Bar Chart Down	~<space> <xstart> <ystart> <length1> <length2>	7E 20 <xstart> <ystart> <length1> <length2>
Draw Bar Chart Right	~! <xstart> <ystart> <length1> <length2>	7E 21 <xstart> <ystart> <length1> <length2>
Draw Bar Chart Left	~" <xstart> <ystart> <length1> <length2>	7E 22 <xstart> <ystart> <length1> <length2>

Table D-3. Remote Commands (cont.)
(Hazeltine 1500 Emulation)

REMOTE COMMANDS	ASCII	HEX
Screen Control Commands		
Execute Stored Screen	~<DLE> <screen #>	7E 10 <screen #>
Receive Stored Screen	~<RS> <screen #>	7E 1E <screen #>
Transmit Stored Screen	~<ESC> <screen #>	7E 1B <screen #>
Jump to Stored Screen	~$ <screen #>	7E 24 <screen #>
Copy Screen Program	~, <source #><dest. #>	7E 2C <source #><dest. #>
Touch Screen Commands [1]		
Change to Normal Mode	~ : <NUL>	7E 3A 0
Change to Touch Screen Mode	~ : <SOH>	7E 3A 1
Set Programmable Touch Screen Zone	~ 9 <zone> <screen>	7E 39 <zone> <screen>
Define Zone or Zones	~ ; <upper left zone> <lower right zone> <code>	7E 3B <upper left zone> <lower right zone> <code>
Additional Commands		
Pause	~ # <time>	7E 23 <time>
Return Password	~ %	7E 25
Plot Point	~0XY	7E 30 X Y
Unplot Point	~1XY	7E 31 X Y

NOTES:

1 For more information on touch screen, see Section 4.2.

Table D-4. Remote Commands
(ANSI Emulation)

Control Characters

00 - ignored
07 - ring bell
08 - move cursor left 1 position
09 - go to next tab stop
0A - linefeed or new line
0B - same as 0A
0C - same as 0A
0D - move cursor to left margin of current line (carriage return)
18 - cancel current ESC sequence
1A - same as 18
1B - ESC

Configuration Commands [2,3]

ESC [? 7 h	- enable autowrap
ESC [? 25 h	- cursor on
ESC [? 7 l	- disable autowrap
ESC [? 25 l	- cursor off
ESC [2 h	- lock keyboard
ESC [2 l	- unlock keyboard
ESC [20 h	- enable auto line-feed
ESC [20 l	- disable auto line-feed
ESC [= 1 h	- cursor on
ESC [= 2 h	- scrolling on
ESC [= 3 h	- treat tab as ANSI tab
ESC [= 1 l	- cursor off
ESC [= 2 l	- scrolling off
ESC [= 3 l	- treat tab as Hazeltine tab
ESC [= 5 h	- enable printer port
ESC [= 5 l	- disable printer port
ESC [= 4 h	- enable screen display
ESC [= 4 l	- disable screen display

Table D-4. Remote Commands (cont.)
(ANSI Emulation)

Attribute Commands [1]

ESC [1 ;attr1;attr2 p	- change character attributes
ESC [m	- attributes off
ESC [0 m	- attributes off
ESC [1 m	- highlight on
ESC [4 m	- underline
ESC [5 m	- blink
ESC [7 m	- reverse video
ESC [22 m	- highlight off
ESC [24 m	- underline disable
ESC [25 m	- blink disable
ESC [27 m	- reverse video off
ESC [50 m	- select regular character set
ESC [51 m	- select double-high characters
ESC [52 m	- select quad-sized characters
ESC [53 m	- select process control symbols
ESC [54 m	- select double-wide characters
ESC [55 m	- select double-size characters
ESC [56 m	- select quad-sized characters
ESC [57 m	- select utility graphics

Table D-4. Remote Commands (cont.)
(ANSI Emulation)

Cursor Movement Commands

ESC [pn A	- cursor up pn lines
ESC [pn B	- cursor down pn lines without scroll
ESC [pn C	- cursor right pn characters
ESC [pn D	- cursor left pn characters
ESC [y;x H	- cursor to position x,y
ESC [H	- cursor home (1,1)
ESC [y;x f	- cursor to position x,y
ESC [f	- cursor home (1,1)
ESC D	- cursor down with scroll
ESC M	- cursor up with scroll
ESC E	- cursor to beginning of next line with scroll
ESC 7	- save cursor and attributes
ESC 8	- restore cursor and attributes

Tab Stop Commands [4]

ESC H	- set tab stop at current column
ESC [g	- clear tab stop at current column
ESC [0 g	- clear tab stop at current column
ESC [3 g	- clear all tab stops

Clear Commands

ESC [pn X	- clear pn characters on current line with background spaces
ESC [K	- clear to end of line with background spaces
ESC [? K	- clear to end of line with background spaces
ESC [0 K	- clear to end of line with background spaces
ESC [? 0 K	- clear to end of line with background spaces
ESC [1 K	- clear to beginning of line with background spaces
ESC [? 1 K	- clear to beginning of line with background spaces
ESC [2 K	- clear entire line with background spaces
ESC [? 2 K	- clear entire line with background spaces
ESC [J	- clear to end of screen with background spaces
ESC [? J	- clear to end of screen with background spaces
ESC [0 J	- clear to end of screen with background spaces
ESC [? 0 J	- clear to end of screen with background spaces
ESC [1 J	- clear to beginning of screen with background spaces
ESC [? 1 J	- clear to beginning of screen with background spaces
ESC [2 J	- clear entire screen with background spaces
ESC [? 2 J	- clear entire screen with background spaces

Table D-4. Remote Commands (cont.)
(ANSI Emulation)

Insert/Delete Commands

ESC [pn L	- insert pn blank line(s) at current cursor position
ESC [pn M	- delete pn line(s) from cursor position
ESC [pn @	- insert pn space(s) in line at cursor position
ESC [pn P	- delete pn character(s) from line at cursor position

Report Commands

ESC [5 n	- device status report device ok returns - ESC [0 n device not ok returns - ESC [3 n
ESC [6 n	- report cursor x,y position, returns - ESC [y;xR
ESC [c	- return options
ESC [0 c	- return options, returns - ESC [? 1;0c

Screen Control Commands

ESC [12;<screen #>p	- execute stored screen
ESC [14;<screen #>p	- receive stored screen
ESC [13;<screen #>p	- transmit stored screen
ESC [19;<screen #>p	- jump to stored screen
ESC [23;<source #>;<dest. #>p	- copy screen program
ESC [24;<K>;<P1>p	- macro key command

Touch Screen Commands [5]

ESC [41; 0 p	- put workstation in normal mode
ESC [41; 1 p	- put workstation in touch screen mode
ESC [40;<zone>;<screen> p	- set programmable touch screen zone
ESC [42; <upper left zone>; <lower right zone>;<code> p	- define zone or zones

Additional Commands

ESC c	- reset to initial state
ESC b	- unlock keyboard
ESC ‘	- lock keyboard
ESC [18;time p	- pause
ESC [20 p	- return password

Table D-4. Remote Commands (cont.)
(ANSI Emulation)

Draw Commands

ESC [2 ;char;ystrt;xstrt;yend;xend p	- draw box
ESC [3 ;char;ystrt;xstrt;length p	- draw vertical line
ESC [4 ;char;ystrt;xstrt;length p	- draw horizontal line
ESC [5 ;ystrt;xstrt;len1;len2 p	- draw bar chart up
ESC [9 p	- background follows
ESC [10 p	- clear foreground
ESC [11 p	- foreground follows
ESC [15;ystrt;xstrt;len1;len2 p	- draw bar chart down
ESC [16;ystrt;xstrt;len1;len2 p	- draw bar chart right
ESC [17;ystrt;xstrt;len1;len2 p	- draw bar chart left
ESC [25;ycor;xcor p	- plot point
ESC [26;ycor;xcor p	- unplot point

NOTES:

1 Multiple attributes can be selected in a single attribute command:
ESC [50;40;31m

2 Multiple configurations can be specified in a single configuration command.
Example:
ESC [= 1;2;3 h
ESC [? 7;25 h
ESC [2;20 h

3 Configuration options that can be set by both the remote commands and the Configuration Menu are not saved on power-down unless the first configuration menu is entered and exited.

4 Tab stops set/reset with remote commands are not saved on power-down unless the Set Tab Stop menu is entered and exited.

5 For more information on touch screen, see Section 4.2.

The following page displays the 2050 Attribute list.

2050 Attribute List

0 (00H) - attributes off (does not affect colors)
4 (04H) - underscore on
5 (05H) - blink on
7 (07H) - reverse colors
24 (18H) - underscore off
25 (19H) - blink off
30 (1EH) - set character color to black
31 (1FH) - set character color to blue
32 (20H) - set character color to green
33 (21H) - set character color to cyan
34 (22H) - set character color to red
35 (23H) - set character color to magenta
36 (24H) - set character color to yellow
37 (25H) - set character color to white
40 (28H) - set screen color to black
41 (29H) - set screen color to blue
42 (2AH) - set screen color to green
43 (2BH) - set screen color to cyan
44 (2CH) - set screen color to red
45 (2DH) - set screen color to magenta
46 (2EH) - set screen color to yellow
47 (2FH) - set screen color to white
50 (32H) - select regular characters
51 (33H) - select double-high characters
52 (34H) - select quad-size characters
53 (35H) - select process graphics characters
54 (36H) - select double-wide characters
55 (37H) - select double-size characters
56 (38H) - select quad-size characters
57 (39H) - select utility graphics

Table D-5. Attribute Byte 1

Bit Number	Attribute
7 (MSB)	not used
6	not used character color bit 2 (2050)
5	not used character color bit 1 (2050)
4	double wide character color bit 0 (2050)
3	blink
2	underline
1	highlight double-wide (2050)
0 (LSB)	reverse video not used (2050)

The definition of attribute byte No. 2 is shown on the following page in Table D-6.

Table D-6. Attribute Byte 2

Bit Number	Attribute
7 (MSB)	not used
6	not used character field color bit 2 (2050)
5	not used character field color bit 1 (2050)
4	not used character field color bit 0 (2050)
3	not used
2	character set bit 2*
1	character set bit 1*
0 (LSB)	character set bit 0*

*See Table D-6A for settings of bits 2 to 0

Table D-6A. Attribute Byte 2, Bits 2 to 0

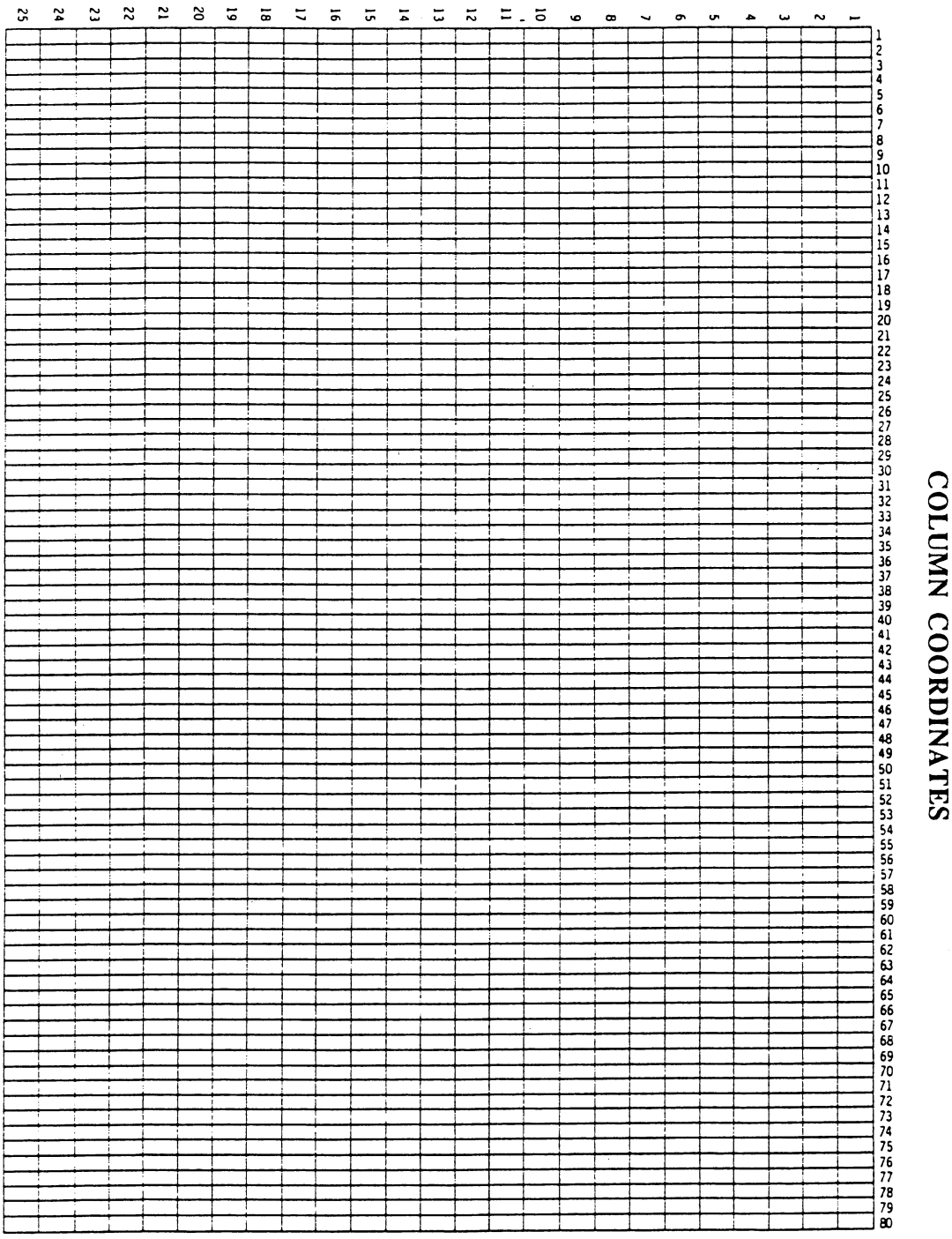
Bit 2	Bit 1	Bit 0	Attributes
0	0	0	regular character
0	0	1	double-high character
0	1	0	quad-size character
0	1	1	process graphic symbols
1	1	1	utility graphics

Table D-6B. Color Select Bits Attribute Byte 1 and 2

Bit 6[1] (red)	Bit 5[1] (green)	Bit 4[1] (blue)	Color
0	0	0	Black
0	0	1	Blue
0	1	0	Green
0	1	1	Cyan
1	0	0	Red
1	0	1	Magenta
1	1	0	Yellow
1	1	1	White

1 In attribute byte 1, these bits select the character color. In attribute byte 2, these bits select the character field color.

ROW COORDINATES



COLUMN COORDINATES

Figure D-2. Video Display Coordinate System (ANSI Emulation)

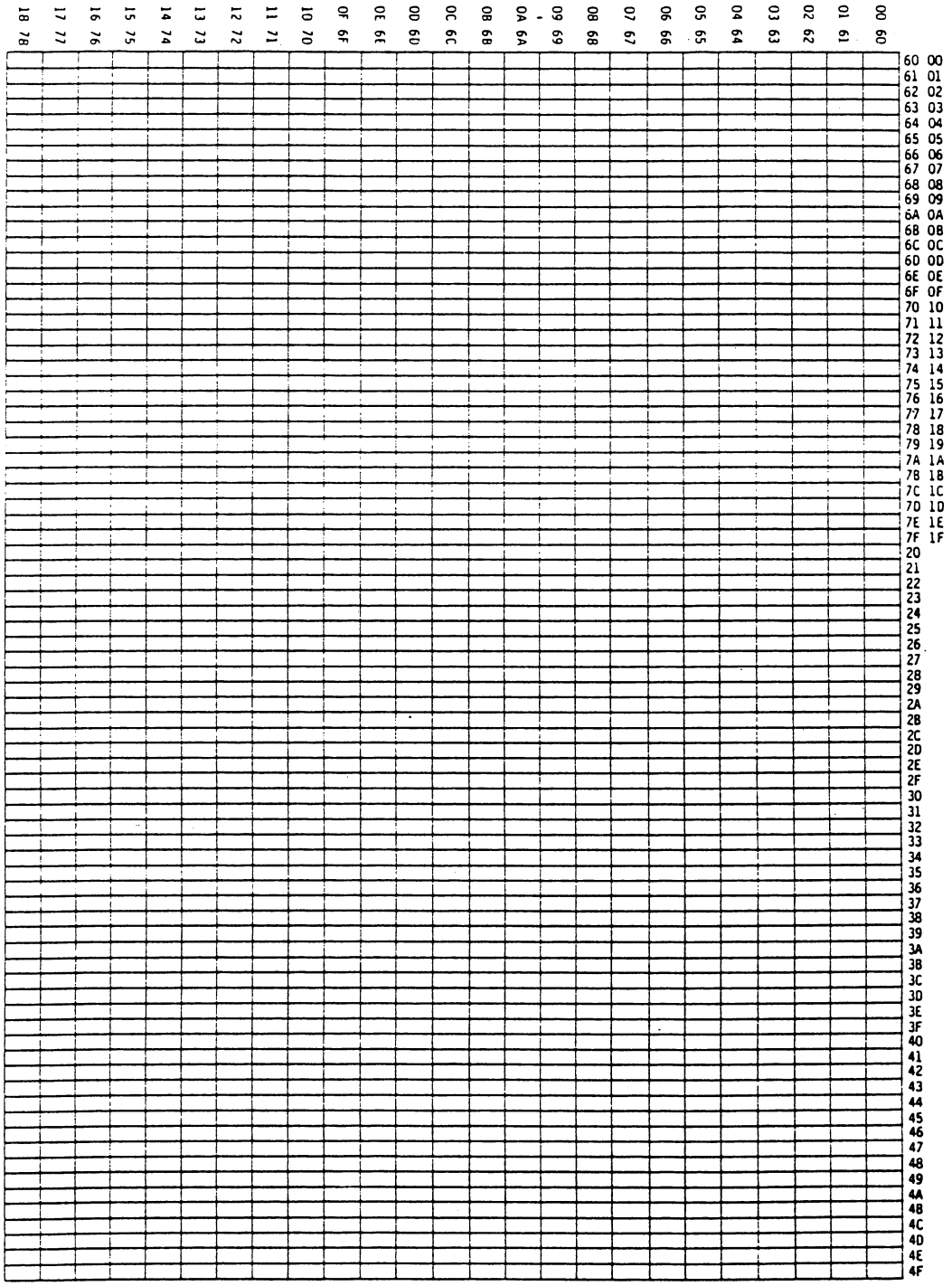


Figure D-3. Video Display Coordinate System (Hazeltine Emulation)

Table D-7. Secondary Serial Port Jumper Settings

Jumper	RS-485	RS-232*
J1	B	B
J2	A	B
J3	A	B
J4	A	B
J5	A	B
J6	A	B
J7	A	B
J8	A	B

* shipping configuration

Table D-8. Primary Serial Port Jumper Settings

Jumper	RS-485	RS-232*
J13	A	B
J14	A	B
J15	A	B
J16	A	B
J17	A	B
J18	A	B
J19	A	B
J20	A	B

* shipping configuration

Table D-9. Signal Termination Jumper Settings*

Secondary Serial Port	Primary Serial Port
RXD: J11, J12 IN CTS: J9, J10 IN	RXD: J23, J24 IN CTS: J21, J22 IN

*shipping configuration - all OUT

Table D-10.
Controller Board Jumper Default Positions

Jumper	Default
J3	OUT*
J5	IN
J6	OUT*
J7	IN
J10	A
J13	IN
J14	IN
J15	B

* permanently wired positions

J10A = OIL Programs in RAM
J10B = OIL programs in EPROM (128K x 8 EPROM installed into U25)

Table D-11. System Configuration Switches*

Switch 1 - Keyboard Type	ON = XT keyboard OFF = AT keyboard
Switch 2 - Touch Screen Option	ON = touch screen OFF = no touch screen
Switch 3 - Location of OIL Programs	ON = OIL programs in EPROM OFF = OIL programs in RAM
Switch 4 and 5 - Matrix Keyboard Type	Both OFF = QWERTY 4 ON, 5 OFF = ABC type
Switches 6 and 7 - Keypad Type	Both OFF = type 0 2005 6 ON, 7 OFF = type 1 Reserved 6 OF, 7 ON = type 2 2050/2060 BOTH ON = type 3 Reserved

*Defaults - all OFF
(if touch screen installed, SW2 default is ON)

Table D-12. Primary Serial Port Pinouts

Pin #	RS-232	RS-485
1	DCD	TXD-
2	RXD	TXD+
3	TXD	RTS-
4	DTR	RTS+
5	GND	GND
6	NC	RXD-
7	RTS	RXD+
8	CTS	CTS+
9	NC	CTS-

Table D-13. Secondary Serial Port Pinouts

Pin #	RS-232	RS-485
1	NC	TXD-
2	RXD	TXD+
3	TXD	RTS-
4	NC	RTS+
5	GND	GND
6	NC	RXD-
7	RTS	RXD+
8	CTS	CTS+
9	NC	CTS-

Table D-14. Matrix Parallel Keyboard Port Pinouts

Pin #	Function	Pin #	Function
1	Column B	14	Row 7
2	Row 1	15	Column 5
3	Column 11	16	Row 8
4	Row 2	17	Column 7
5	Column 10	18	Column 12
6	Row 3	19	Column 6
7	Column 3	20	Column 13
8	Row 4	21	Column 0
9	Column 2	22	No Connect
10	Row 5	23	Numlock Rstr.
11	Column 1	24	Numlock
12	Row 6	25	Caplock Rstr.
13	Column 4	26	Caplock

Table D-15. Parallel Input/Output Port Pinouts

Pin #	Signal	Pin #	Signal
*1	STROBE	10	ACK
*2	DATA0	11	BUSY
*3	DATA1	12	NC
*4	DATA2	13	NC
*5	DATA3	14	NC
*6	DATA4	15	NC
*7	DATA5	16	RESET
*8	DATA6	17	NC
*9	DATA7	*18-25	GND

* Used for input mode

Table D-16. Keyboard Connector Pinouts

Pin #	Signal
1	Clock
2	Data
3	NC
4	GND(SG)
5	+5 VDC
6	GND(FG)

E.1 INTRODUCTION

The 2005 and 2050/2060 Workstations have the same functions and features as the 2000 Workstation, but also has front panel keypads. The 2005 has 20 functions keys (located beneath the screen) and a 34-key keypad (located to the right of the screen). The 2005 front panel is shown in Figure E-1 below. The 2050/2060 front panel has 20 function keys (located beneath the screen) and a 37-key keypad (located at the right of the screen). The 2050/2060 front panel is shown in Figure E-2, on the following page.

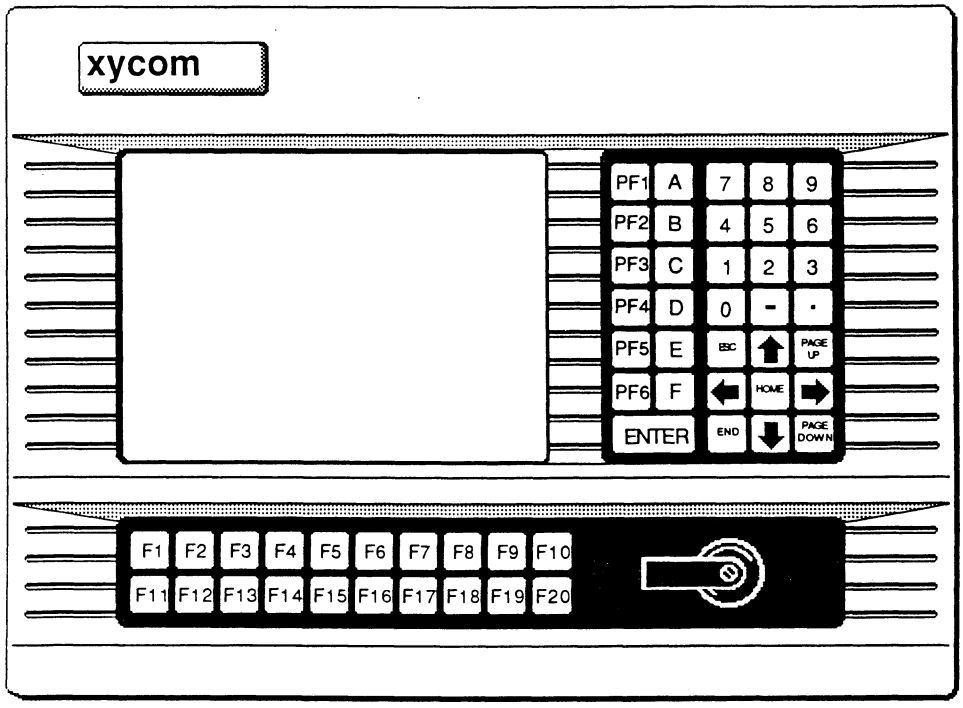


Figure E-1. 2005 Front Panel

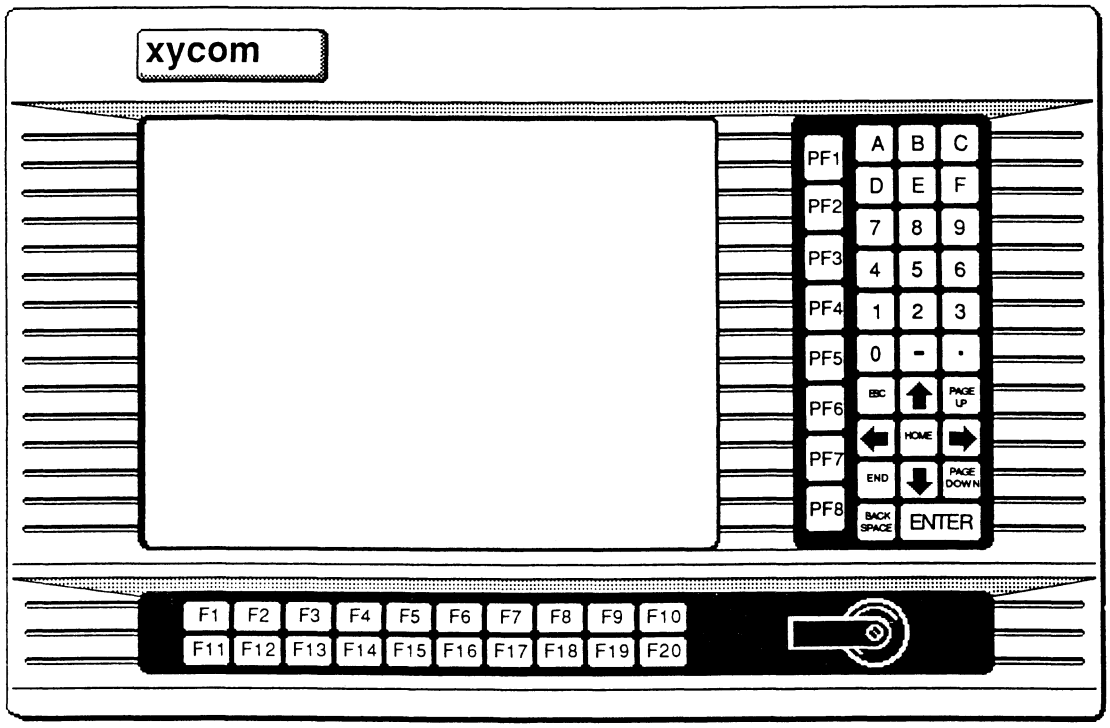


Figure E-2. 2050/2060 Front Panel

E.2 CODES RETURNED FROM KEYPADS

In standard mode, the keypads generate hexadecimal codes when pressed. The codes are shown in Figures E-3 and E-4 on the following page.

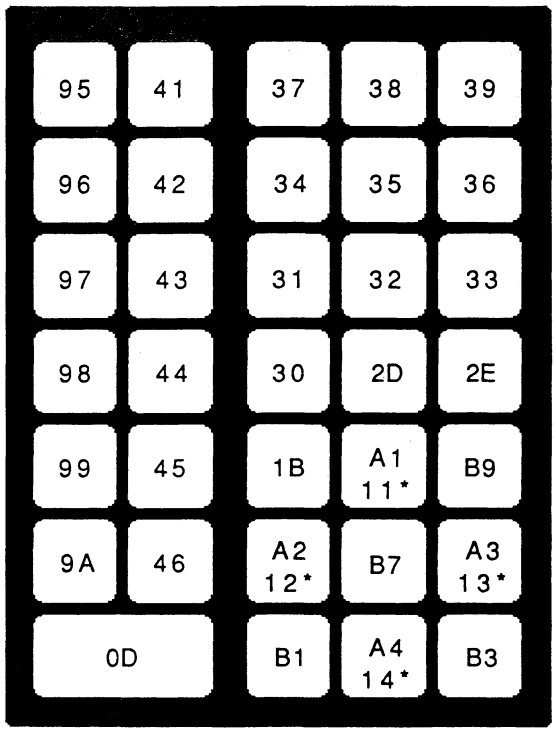


Figure E-3. Hexadecimal Keypad Codes, Standard Mode

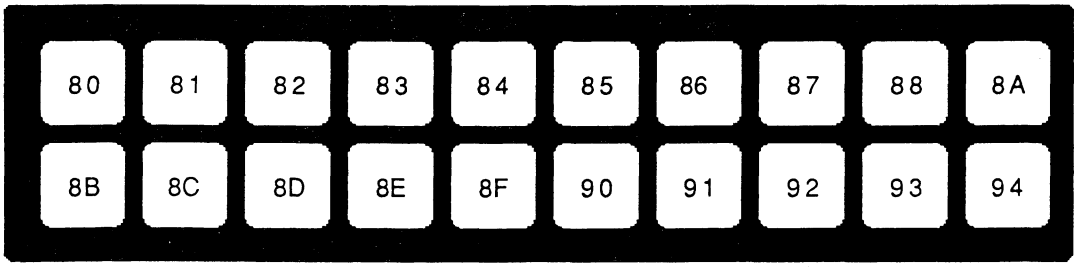


Figure E-4. Hexadecimal Function Key Code, Standard Mode

In application mode, hexadecimal values are returned in Hazeltine mode, and ASCII characters are returned in ANSI mode. See the Figures E-5, E-6, and E-7 for more information

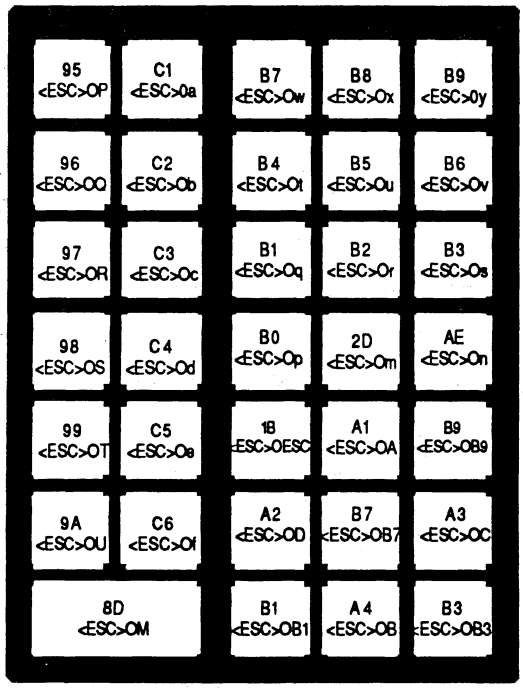


Figure E-5. Hex and ASCII Keypad Codes, Application Mode

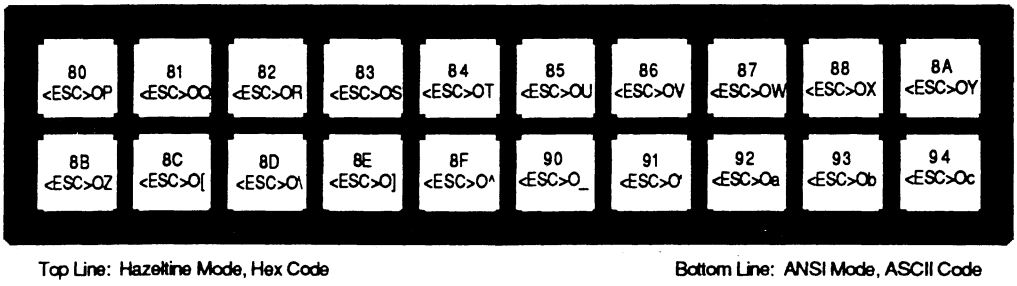


Figure E-6. Hex and ASCII Function Keys Codes, Application Mode

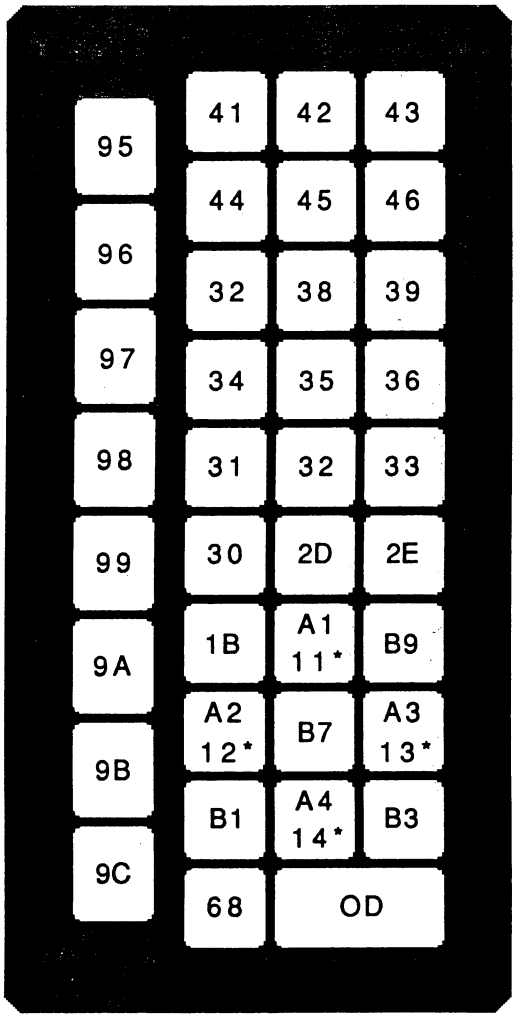


Figure E-7. 2050/2060 Hexadeximal Keypad Codes Standard Mode

E.3 RELEGENDING THE KEYPADS

The function keys (34 through 53) and keys 0 through 5 and 54 and 55 on the keypad are relegendable (see Figures E-8 and E-9 below. Due to the construction of the 2005 and 2050/2060, accessing the keys involves several steps. For information on accessing or changing the keys, consult Xycom.

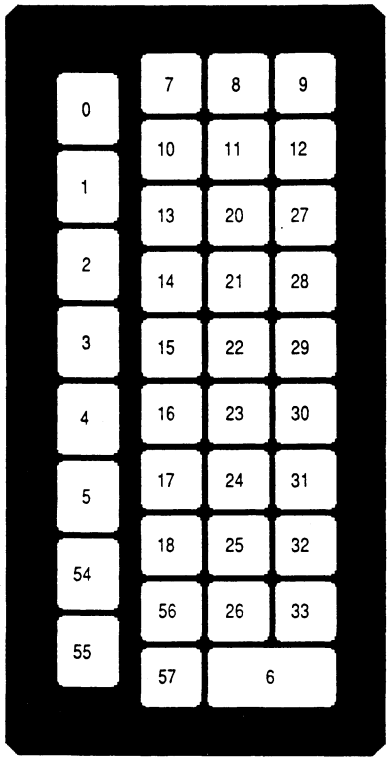


Figure E-8. 2050/2060 Programmable Keypad

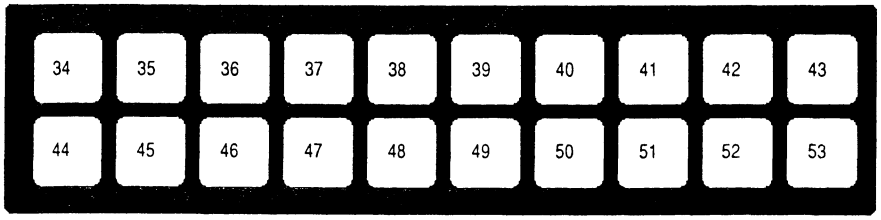


Figure E-9. 2050/2060 Function Keys

In the application mode, hexadecimal values are returned in Hazeltine mode, and ASCII characters are returned in ANSI mode. See Figure E-10 below for more information.

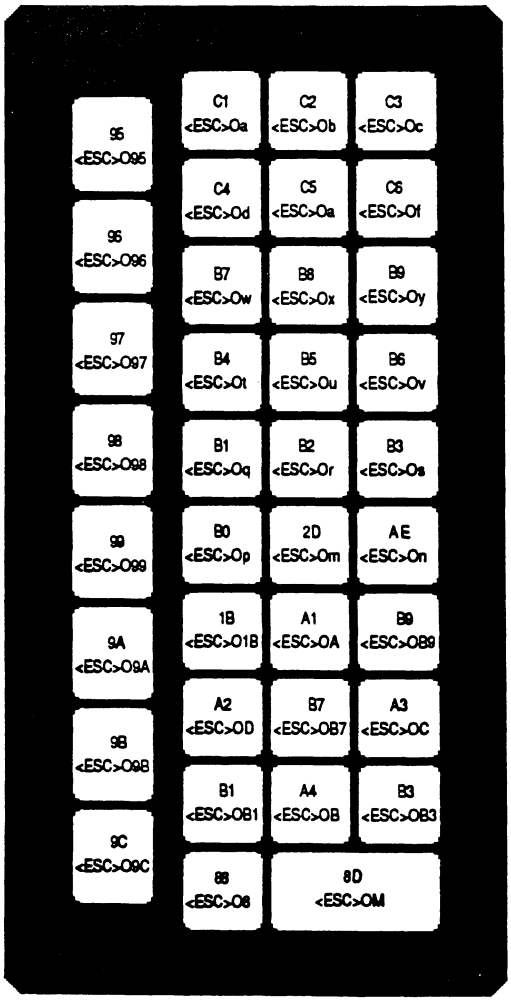


Figure E-10. 2050/2060 Hex and ASCII Keypad Codes, Application Mode

INDEX

NUMERIC

A

B

C

C (Continued)

D

D (Continued)

E

F

G

H

I

J

K

L

M

N

P

P (Continued)

Q

R

S

S (Continued)

T

U

V