

# **GRAPHICS-PLUS**

## **OPERATIONS    MANUAL**



**NORTHWEST DIGITAL SYSTEMS**

**Seattle    Washington**

### WARRANTY

The GRAPHICS-PLUS hardware is warranted against defects in materials and workmanship for a period of 90 days following the date of purchase. This warranty shall be void if the GRAPHICS-PLUS hardware is damaged by improper installation. The responsibility and liability of Northwest Digital Systems shall be limited to repair or replacement of the GRAPHICS-PLUS hardware.

### CUSTOMER SERVICE INFORMATION

Any questions regarding service or repairs should be referred to:

NORTHWEST DIGITAL SYSTEMS  
P.O. Box 15288  
Seattle, Washington 98115  
206-362-6937

GRAPHICS-PLUS model GP-19 Operations Manual  
Part No. 14-00001-02

Copyright 1983 Northwest Digital Systems  
All Rights Reserved

Printed in the United States

Comments about this manual would be appreciated by the authors.  
Comments and suggestions may be mailed to:

Northwest Digital Systems  
P.O. Box 15288  
Seattle, Washington 98115

TEKTRONIX is a registered trademark of Tektronix, Inc.

DEC, VT, VAX, and VMS are registered trademarks of Digital Equipment Corporation.

GRAPHICS-PLUS and GP-19 are registered trademarks of Northwest Digital Systems.

|           |   |      |
|-----------|---|------|
| CHAPTER 1 | INTRODUCTION  |      |
| 1.1       | PRODUCT SUMMARY . . . . .   | 1-1  |
| 1.2       | MANUAL OVERVIEW . . . . .   | 1-3  |
| 1.3       | NOMENCLATURE . . . . .  | 1-3  |
| CHAPTER 2 | SPECIFICATIONS  |      |
| 2.1       | PHYSICAL PACKAGING . . . . .  | 2-1  |
| 2.2       | GRAPHICS MODE SPECIFICATIONS . . . . .                              | 2-1  |
| 2.3       | TEXT MODE SPECIFICATIONS . . . . .                                  | 2-2  |
| 2.4       | GENERAL SPECIFICATIONS . . . . .                                    | 2-2  |
| 2.5       | Z19 COMPATIBILITY . . . . .   | 2-3  |
| 2.6       | OPTIONS . . . . .   | 2-4  |
| CHAPTER 3 | INSTALLATION  |      |
| 3.1       | INSTALLING THE BOARD . . . . .                                      | 3-1  |
| 3.1.1     | GP-19 Installation In Z19, H19 Or WH19 Terminals                    | 3-4  |
| 3.1.2     | GP-19 Installation In Z19-CN, Z19-HN Or H19A<br>Terminals . . . . . | 3-7  |
| 3.1.3     | GP-19-CN Power Supply Installation (Z19-CN Only)                    | 3-9  |
| 3.1.4     | Testing And Adjustments . . . . .                                   | 3-12 |
| 3.1.5     | In Case Of Difficulty . . . . .                                     | 3-14 |
| 3.2       | SWITCH SETTINGS . . . . .   | 3-15 |
| CHAPTER 4 | OPERATION   |      |
| 4.1       | OVERVIEW . . . . .  | 4-1  |
| 4.2       | LOCAL FUNCTION KEYS . . . . .                                       | 4-2  |
| 4.3       | SET-UP MODE . . . . .   | 4-4  |
| 4.4       | TEXT MODE FORMATS . . . . .   | 4-6  |
| 4.5       | OFF-SCREEN MEMORY . . . . .   | 4-7  |
| 4.5.1     | Scrolling Backward . . . . .  | 4-8  |
| 4.5.2     | Scrolling Forward . . . . .   | 4-8  |
| 4.5.3     | Scrolling Speed . . . . .   | 4-9  |
| 4.5.4     | Inserting Lines In Display Memory . . . . .                         | 4-9  |
| 4.5.5     | Clearing Display Memory . . . . .                                   | 4-10 |
| 4.6       | TEKTRONIX MODE . . . . .  | 4-11 |
| 4.6.1     | Alpha State Operation . . . . .                                     | 4-12 |
| 4.6.1.1   | Interaction With Graph State . . . . .                              | 4-13 |
| 4.6.1.2   | Line Length . . . . .   | 4-13 |
| 4.6.1.3   | Status/position Readback . . . . .                                  | 4-13 |
| 4.6.2     | Graph State Operation . . . . .                                     | 4-14 |
| 4.6.2.1   | Specifying Vector End Points . . . . .                              | 4-15 |
| 4.6.2.2   | Specifying The Vector Type . . . . .                                | 4-17 |
| 4.6.2.3   | Area Fill And Erase Vectors . . . . .                               | 4-18 |

|          |   |      |
|----------|---|------|
| 4.6.2.4  | Status/position Readback . . . . .                | 4-18 |
| 4.6.3    | Graphics Input (GIN) State Operation . . . . .    | 4-19 |
| 4.6.3.1  | Pre-Positioning The GIN Cursor . . . . .          | 4-21 |
| 4.6.4    | Selective Erase Operations . . . . .              | 4-21 |
| 4.6.5    | Control Characters And Escape Sequences . . . . . | 4-22 |
| 4.7      | EASY-ENTRY GRAPHICS MODE . . . . .                | 4-25 |
| 4.8      | GRAPHICS SCROLLING . . . . .                      | 4-28 |
| 4.9      | TRANSPARENT MODE . . . . .                        | 4-28 |
| 4.10     | PROGRAMMABLE KEYS . . . . .                       | 4-29 |
| 4.11     | REVERSE SCREEN STATE . . . . .                    | 4-31 |
| 4.12     | ADDITIONAL NEW FEATURES . . . . .                 | 4-31 |
| 4.12.1   | Scroll Key . . . . .                              | 4-31 |
| 4.12.2   | Programmable Tab Stops . . . . .                  | 4-32 |
| 4.12.3   | Back Tab Sequence . . . . .                       | 4-33 |
| 4.12.4   | Smart Shift Key . . . . .                         | 4-33 |
| 4.12.5   | Smart Auto Wrap . . . . .                         | 4-33 |
| 4.12.6   | Screen Timeout . . . . .                          | 4-34 |
| 4.12.7   | Transmit Cursor Line . . . . .                    | 4-34 |
| 4.12.8   | Transmission Speed Limiting . . . . .             | 4-35 |
| 4.12.9   | Definable Scrolling Region . . . . .              | 4-35 |
| 4.12.9.1 | Relative Vs. Absolute Origin Mode . . . . .       | 4-36 |
| 4.12.10  | EDT Compatibility . . . . .                       | 4-37 |
| 4.12.11  | VT100 Character Set . . . . .                     | 4-38 |
| 4.12.12  | Status Line Addressing . . . . .                  | 4-39 |
| 4.12.13  | Output Synchronization . . . . .                  | 4-39 |
| 4.12.14  | Variable Length Key Click . . . . .               | 4-40 |
| 4.13     | ESCAPE SEQUENCES . . . . .                        | 4-40 |
| 4.13.1   | ANSI Compatible Escape Sequences . . . . .        | 4-40 |
| 4.13.1.1 | EDT Compatibility . . . . .                       | 4-41 |
| 4.13.2   | Global Escape Sequences . . . . .                 | 4-42 |
| 4.13.3   | Other New Escape Sequences . . . . .              | 4-42 |
| 4.14     | INPUT AND OUTPUT BUFFERS . . . . .                | 4-43 |

CHAPTER 5 TROUBLESHOOTING

|     |                                      |     |
|-----|--------------------------------------|-----|
| 5.1 | TROUBLESHOOTING PROCEDURES . . . . . | 5-1 |
| 5.2 | USING THE SELF-DIAGNOSTICS . . . . . | 5-2 |

CHAPTER 6 CIRCUIT DESCRIPTION

|       |  |     |
|-------|--|-----|
| 6.1   | INTERCONNECTION WITH THE Z19 . . . . . | 6-1 |
| 6.2   | POWER SUPPLY . . . . .                 | 6-2 |
| 6.3   | CLOCK LOGIC . . . . .                  | 6-2 |
| 6.3.1 | Oscillators . . . . .                  | 6-2 |
| 6.3.2 | Timing Signal Generation . . . . .     | 6-3 |
| 6.4   | CPU AND SYSTEM CONTROL LOGIC . . . . . | 6-3 |
| 6.4.1 | Memory And I/O Port Mapping . . . . .  | 6-4 |
| 6.5   | CRT CONTROLLER . . . . .               | 6-6 |

|       |   |      |
|-------|---|------|
| 6.5.1 | Read/write Logic . . . . .                              | 6-6  |
| 6.5.2 | Display Control Signals . . . . .                       | 6-7  |
| 6.5.3 | Vertical Retrace Interrupt . . . . .                    | 6-8  |
| 6.6   | MEMORY AND MEMORY ARBITRATION LOGIC . . . . .           | 6-9  |
| 6.6.1 | Dynamic Memory Operation . . . . .                      | 6-9  |
| 6.6.2 | Memory Access Arbitration And Synchronization . . . . . | 6-10 |
| 6.6.3 | Address Generation . . . . .                            | 6-12 |
| 6.6.4 | Refresh Logic . . . . .                                 | 6-13 |
| 6.7   | CHARACTER AND DOT GENERATION . . . . .                  | 6-14 |
| 6.7.1 | Character Generation . . . . .                          | 6-15 |
| 6.7.2 | Graphics Dot Generation . . . . .                       | 6-16 |
| 6.7.3 | Blanking, Reverse Video, And Reverse Screen . . . . .   | 6-16 |

APPENDIX A            ESCAPE SEQUENCES AND CONTROL CHARACTERS

|      |   |      |
|------|---|------|
| A.1  | CONTROL CHARACTERS . . . . .  | A-1  |
| A.2  | ESCAPE SEQUENCE DEFINITIONS . . . . .                                       | A-2  |
| A.3  | TERMINAL MODE CONTROL . . . . .   | A-3  |
| A.4  | KEYPAD MODE CONTROL . . . . .   | A-4  |
| A.5  | ESCAPE SEQUENCE REPORTS SENT BY THE TERMINAL . . . . .                      | A-4  |
| A.6  | ESCAPE SEQUENCES SENT BY THE KEYPAD AND FUNCTION KEYS . . . . .             | A-5  |
| A.7  | CURSOR POSITION OPERATIONS . . . . .  | A-6  |
| A.8  | SCREEN ERASURE OPERATIONS . . . . .   | A-7  |
| A.9  | SCREEN EDITING OPERATIONS . . . . .   | A-7  |
| A.10 | TAB-SETTING OPERATIONS . . . . .  | A-8  |
| A.11 | SCROLLING OPERATIONS . . . . .  | A-8  |
| A.12 | SCREEN FORMATTING . . . . .   | A-9  |
| A.13 | GRAPHIC RENDITION SEQUENCES . . . . .                                       | A-9  |
| A.14 | PROGRAMMABLE FUNCTION KEYS . . . . .  | A-10 |
| A.15 | EXTERNAL ROM SEQUENCES . . . . .  | A-11 |
| A.16 | CONFIGURATION SEQUENCES . . . . .   | A-11 |
| A.17 | TEKTRONIX MODE ESCAPE SEQUENCES . . . . .                                   | A-12 |
| A.18 | MISCELLANEOUS ITEMS . . . . .   | A-13 |
| A.19 | IGNORED VT100 ESCAPE SEQUENCES . . . . .                                    | A-13 |
| A.20 | SUMMARY OF SEQUENCES WITH MULTIPLE PARAMETERS . . . . .                     | A-14 |
| A.21 | SUMMARY OF SEQUENCES RECOGNIZED IN ZENITH ESCAPE MODE . . . . .             | A-17 |
| A.22 | SUMMARY OF SEQUENCES RECOGNIZED IN ANSI ESCAPE MODE . . . . .               | A-20 |
| A.23 | SUMMARY OF SEQUENCES RECOGNIZED IN TEKTRONIX AND EASY-ENTRY MODES . . . . . | A-24 |

APPENDIX B            VECTOR DRAWING TIMES

|            |                              |
|------------|------------------------------|
| APPENDIX C | COMPOSITE VIDEO GENERATOR    |
| APPENDIX D | AUXILIARY I/O CONNECTOR      |
| APPENDIX E | VT100 GRAPHICS CHARACTER SET |
| APPENDIX F | ASCII CHARACTERS             |
| APPENDIX G | OPTIONAL SOFTWARE            |
| APPENDIX H | CIRCUIT DIAGRAMS             |
| APPENDIX I | PARTS LIST                   |

## CHAPTER 1

### INTRODUCTION

#### 1.1 PRODUCT SUMMARY

GRAPHICS-PLUS is a field-installable enhancement board for the Zenith Z19 video terminal. Addition of GRAPHICS-PLUS to a Z19 provides Tektronix 4010 compatible vector graphics, 80/132 column and 24/49 line display formats, off-screen scrolling memory, programmable function keys, menu driven Set-up mode, and numerous other features.

Graphics capabilities include the following:

- \* 512 horizontal by 250 vertical resolution
- \* Tektronix 4006, 4010 graphic and alpha mode compatibility
- \* Tektronix Graphics Input mode using the terminal's keypad
- \* White-on-black and black-on-white vectors
- \* Selective erase and area erase vectors
- \* Area fill vectors
- \* Easy-Entry graphics mode allowing decimal specification of coordinates
- \* Local function keys for screen erase and enter/exit Tektronix mode



## INTRODUCTION

Text modes include the following formats:

- \* 80 column by 24 lines (plus 25th status line)
- \* 132 column by 24 lines (plus 25th status line)
- \* 80 column by 49 lines (plus 50th status line)
- \* 132 column by 49 lines (plus 50th status line)

Other significant enhancements provided by GRAPHICS-PLUS include:

- \* Seven pages of off-screen memory
- \* Local function keys for scrolling through memory
- \* CRT-saver timeout feature
- \* True reverse screen capability
- \* VT100 style scroll key
- \* Definable scrolling region
- \* Menu driven Set-up mode
- \* Programmable function keys with 512 bytes total memory
- \* Transparent mode for displaying all received characters including control codes
- \* Fully programmable tab stops
- \* Self diagnostics for checking all RAMS and ROMS

All GRAPHICS-PLUS logic is contained on a single circuit board that can be quickly installed using only a screwdriver.

## INTRODUCTION

### 1.2 MANUAL OVERVIEW

This manual contains all information necessary to install and operate the GRAPHICS-PLUS enhancement board. Before attempting to install GRAPHICS-PLUS the user should read Chapter 3; all necessary instructions for installation and testing can be found in that chapter. Detailed operating information, both for the terminal user and the programmer, is in Chapter 4. For the user who requires a detailed understanding of the device hardware, Chapter 6 provides a complete circuit description. Troubleshooting information can be found in Chapter 5.

### 1.3 NOMENCLATURE

References in the text of this manual to a Z19 terminal refer to Zenith Z19, Z19-CN, and Z19-HN (or Heath H19, H19A, and WH19) terminals interchangeably unless explicitly stated otherwise. References to "the terminal" refer to any one of the above models equipped with a GRAPHICS-PLUS enhancement board.

The term "main keyboard" refers to all keys except those located in the numeric keypad. "Numeric keypad" refers to the group of twelve keys located on the right side of the keyboard. Keys are referred to in this text just as they are labelled on the keyboard, with the label spelled all in capitals. For example, the words SHIFT and CTRL refer to the keys of the same name on the terminal keyboard. Note that neither of these keys actually produces a character when struck; instead they modify the codes produced by other keys struck when one or both of them are held down. Thus if reference is made to the effect of typing CTRL-F1, it refers to striking the F1 key while holding down the CTRL key. Likewise, CTRL-SHIFT-F1 refers to striking the F1 key while holding down both CTRL and SHIFT. The word ESC, all in capitals, also refers to the key of the same name on the terminal keyboard and is an abbreviation for the ASCII character named ESCAPE. This key, in contrast to SHIFT and CTRL, actually produces a code when struck (the ASCII ESCAPE character).

Characters received or transmitted by the terminal are referred to by their standard ASCII names. Control characters are represented by two or three character symbolic names, spelled in capitals. CR, for instance, is the character produced by striking RETURN. Tables showing the hexadecimal values of all ASCII characters and the codes associated with all ASCII control codes can be found in Appendix F.

Escape sequences are printed in one of two different ways. If they appear in line with text they are surrounded by parenthesis to separate them from the adjoining text (ESC d). The parenthesis are not part of the sequence. If the sequence is printed on a separate line parenthesis

## INTRODUCTION

are not included. For purposes of clarity the characters in an escape sequence are separated by spaces. These spaces are not actually part of the sequence and are never to be included when typing an escape sequence.

The term "global escape sequence" refers to a new class of escape sequences implemented by GRAPHICS-PLUS. These escape sequences are recognized by the terminal without regard to whether the terminal is currently in ZENITH escape mode or ANSI escape mode. Detailed information on escape sequences can be found in Section 4.13 and Appendix A.

## CHAPTER 2

### SPECIFICATIONS

#### 2.1 PHYSICAL PACKAGING

GRAPHICS-PLUS is packaged on a single 11" by 9.75" printed circuit card which mounts in the forward card position of a Z19 or Z19-CN terminal. The circuit boards supplied for different models of the terminal are electrically different and are not interchangeable. Power consumption is about ten watts. A separate power supply is included with boards for Z19-CN terminals.

#### 2.2 GRAPHICS MODE SPECIFICATIONS

The terminal supports a superset of the features and communications protocol of the Tektronix model 4010 terminal.

##### Graphics format:

1. 512 dots wide by 250 dots high display format
2. 8.25 inches wide by 6.5 inches high display area
3. 7x7 alpha character cell
4. 73 character x 35 line alpha format

##### Graphics performance:

1. Full length vectors can be continuously drawn at 1200 baud
2. 25 millisecond longest vector drawing time
3. Input buffering and XON/XOFF protocol support allows operation at 9600 baud
4. 96 ms maximum screen erase time

Complete information on vector drawing times can be found in Appendix B.

## SPECIFICATIONS

### Graphics protocol extensions:

1. Selective vector and character erase
2. Area fill and area erase
3. Easy-Entry decimal notation protocol
4. Horizontal scrolling of selected region

### 2.3 TEXT MODE SPECIFICATIONS

1. 16,384 characters of scrolling display memory (203 lines in 80 column mode, 123 lines in 132 column mode)
2. Local function keys are provided to:
  1. Scroll forward a page
  2. Scroll backward a page
  3. Scroll forward a line
  4. Scroll backward a line
  5. Go to beginning of display memory
  6. Go to end of display memory
  7. Erase all of display memory
  8. Erase from current cursor position to end of display memory
  9. Erase the screen
  10. Erase from the current cursor position to the bottom of the screen

### 2.4 GENERAL SPECIFICATIONS

1. Tab stop positions are fully programmable by the user and can be set at any column. The programmable tab stops can be set without disturbing the normal default tab stops (every eight columns) that are in effect when programmable tab stops are not enabled.
2. Sixteen function keys may be programmed to transmit sequences of ASCII characters. Each function key may be programmed with a sequence of up to 128 characters, with the total of all sequences not to exceed 512 characters. All ASCII characters, including control codes, may be used in these sequences.

## SPECIFICATIONS

3. The terminal has a true reverse screen capability. When in reverse screen state all characters are displayed as dark characters on a white background, giving a "printed page" appearance.
4. The terminal has a 1024 character input buffer and a 128 character output buffer. XON/XOFF protocol is supported for both input and output.

### NOTE

GRAPHICS-PLUS does not support operation with a 50 HZ vertical sync rate. When GRAPHICS-PLUS is installed switch S402 section 7 on the Z19 logic board, which formerly selected 50 HZ or 60 HZ vertical sync rate, is assigned a new function (see Section 3.2) and 60 HZ vertical sync is always used. Operation from a 50 HZ power source may result in a slightly unstable screen image.

### 2.5 Z19 COMPATIBILITY

GRAPHICS-PLUS has been designed in such a way as to preserve all the original functions of the Z19 terminal while adding additional features. All escape sequences recognized by the Z19 are recognized by GRAPHICS-PLUS and executed in a compatible manner with one exception. The "reset to power up configuration" sequences (ESC z ESC c ESC [ z) are executed by the the Z19 in less than one character time; i.e., the reset is effectively instantaneous. When GRAPHICS-PLUS is installed the reset operations takes longer, a maximum of 350 msec, which means that characters sent immediately after a reset escape sequence may be lost. Application programs which use a reset escape sequence should either pause for at least 350 msec, or send a sufficient number of NUL characters, before sending additional characters to the terminal.

## SPECIFICATIONS

### 2.6 OPTIONS

1. A socket is provided for connecting an optional composite video generator. Complete information on this option can be found in Appendix C.
2. A socket is provided for installing an auxiliary I/O card. This card may contain additional program ROM space, RAM and I/O devices. Hardware details for this option can be found in Appendix D.

## CHAPTER 3

### INSTALLATION

#### 3.1 INSTALLING THE BOARD

Installation Time: 15 minutes

Required Tools:     Slotted screwdriver  
                          Pliers (Z19-CN terminals only)

Make sure that the model of the terminal agrees with that specified in the "Unpacking Instructions". Terminal model designations can be found on the back of the terminal and are most commonly one of the following: Z19, Z19-CN, Z19-HN, H19, H19A, or WH19.

If the model of the terminal does not agree with that specified in the "Unpacking Instructions" contact Northwest Digital Systems. Do NOT attempt to install the GP-19 enhancement board into a model of the terminal for which it is not intended.

Installation of the GP-19 enhancement board requires moving two ICs (or "chips"). Those having little or no technical expertise should read the "Notes to Non-technical Installers" on the following page for some hints regarding this step. If any problems are encountered in transferring ICs to the GP-19 logic board (such as damaged pins) please contact Northwest Digital Systems.

In the event any difficulties or questions arise, please contact Northwest Digital Systems for assistance.



## INSTALLATION

### Notes To Non-Technical Installers

- \* Component locations on the GP-19 logic board can be determined from the component layout diagram in Appendix H, page H-9 of this manual. Component locations are also identified by small numbers printed on the GP-19 logic board just above each IC.
- \* All ICs on the Zenith logic board are in sockets. A small screwdriver should be used to remove the specified ICs from their sockets. This can be accomplished by inserting the screwdriver between the IC and its socket and gently prying up. Since the pins on the ICs are delicate and are easily damaged, it is best to "work" the IC out of the socket by prying a little from each end. Be careful not to mistakenly pry the socket off of the logic board instead of removing the IC from its socket. It may prove helpful to look at location U27 on the GP-19 logic board to see what a socket without an IC looks like. Avoid scratching the logic board with the screwdriver while removing the ICs.
- \* When inserting ICs into the GP-19 logic board, it is best to have the logic board on a flat surface. Be careful to insert all the pins and press the IC firmly into the new socket. Proper insertion may require substantial force. The IC should be pressed all the way against the socket. Observe other ICs on the GP-19 logic board to determine how far the IC should be inserted. Be aware that occasionally pins bend up underneath ICs - a particularly difficult fault to detect.

## INSTALLATION

### INSTALLATION INSTRUCTIONS

1. Verify that the Z19 terminal is operational and functions normally.
2. Verify that all parts listed in the "Unpacking Instructions" are present. Identify each of the following components:
  - \* GP-19 Logic board
  - \* Processor interconnect cable  
(A 7 inch 40 conductor flat ribbon cable assembly.)
  - \* Power interconnect cable  
(A 5 or 6 wire multi-colored 6 inch cable assembly.)
  - \* Plastic keyboard label
3. Unplug the terminal from the AC line.
4. Gain access to the inside of the terminal by removing the cover. For instructions on cabinet removal refer to the "SET-UP" section in the VIDEO TERMINAL OPERATION MANUAL supplied by Zenith.
5. Locate the 9 inch by 11 inch printed circuit card standing upright near the back of the terminal - hereafter referred to as the "Zenith logic board".

For Z19-CN, Z19-HN and H19A terminals, skip Section 3.1.1 and proceed to Section 3.1.2.

## INSTALLATION

### 3.1.1 GP-19 Installation In Z19, H19 Or WH19 Terminals

To complete steps 1-3, installers with little or no technical expertise should refer to the "Notes to Non-Technical Installers" in Section 3.1.

1. Transfer the 40 pin IC from location U421 on the Zenith Logic board to location U27 near the top center of the GP-19 Logic board. U421 on the Zenith Logic board is positioned approximately 4 inches from the top of the Logic board and just to the right of center. Make sure the IC is inserted with pin 1 toward the right side of the GP-19 Logic board. Note: The serial number appears on the "left" side of the GP-19 Logic board, and the metal heatsink is located on the "top" of the board. Pin 1 of the IC is identified in Figure 3.1.



Figure 3.1

2. Transfer the 40 pin IC from location U461 on the Zenith Logic board to location U50 near the top left of the GP-19 Logic board. U461 is approximately 4 inches to the left of the IC removed in step 1. Make sure the IC is inserted with pin 1 toward the right side of the GP-19 Logic board. (If the GP-19 was shipped with an IC in location U50 then omit this step.)
3. 

\*\*\*\*\* IMPORTANT \*\*\*\*\*

Recheck to make sure the ICs transferred to the GP-19 Logic board have pin 1 correctly positioned. (See Figure 3.1.) Pin 1 should be on the right end of the IC as viewed from the component side of the GP-19 Logic board. The writing on the ICs should appear upside-down.
4. Inspect location U423 on the Zenith Logic board. U423 is approximately 2 inches below the IC removed in step 1. If there is an IC in this location it must be removed. (Only the earliest versions of the terminal will have an IC in this location.) Although the IC is not used by the GP-19, it should not be discarded in case the GP-19 is ever removed from the terminal.
5. Remove the foam strip protecting the pins of the 40 pin DIP-plug connected to the processor interconnect cable supplied with the GP-19.

## INSTALLATION

6. Insert the 40 pin DIP-plug connector of the processor interconnect cable into the socket on the Zenith logic board that was vacated in step 1 (U421). This plug should be inserted so that the attached cable enters the plug from the top of the terminal. Make sure all 40 pins are inserted into the socket and that no pins bend up underneath the DIP-plug.
7. To prevent the ribbon cable from interfering with the installation of the GP-19 logic board fold it backwards over the top of the Zenith logic board.
8. Disconnect the multi-colored video/power cable connected to the upper left corner of the Zenith logic board.
9. Connect either end of the power interconnect cable supplied with the GP-19 to the connector in the upper left corner of the Zenith logic board.

### NOTE

This cable utilizes a "keyed" connector. Early versions of the Z19 terminal did not key this connector. If the terminal is one of the earlier versions it is recommended that the proper pin (fourth pin from the left as viewed from the front of the terminal) be removed from connector P401. Alternatively, the plastic keying pin in the power interconnect cable may be removed.

10. Remove the two 6-32 screws from the black plastic inserts in the metal heatsink at the top of the GP-19 logic board.
11. Place the GP-19 logic board just in front of the Zenith logic board with the component side of the board towards the front of the terminal. The GP-19 logic board should rest in the two slots at the bottom of the terminal.
12. Secure the GP-19 metal heatsink to the upright mounting brackets using the two screws removed in step 10.
13. Fold the 40 conductor processor interconnect cable over the front of the GP-19 board and connect it to connector P2 on the GP-19 logic board. The connectors should be pressed firmly together. Make sure that the connectors are not "offset" with only one row of pins connected.

## INSTALLATION

14. Connect the power interconnect cable to connector P8 on the GP-19 logic board. This connector is keyed and will connect only if properly oriented.
15. Connect the multi-colored video/power cable from the terminal (disconnected in step 8) to connector P6 on the GP-19 logic board. Route this cable so that it does not rest on the grey rubber flyback transformer.

This completes the installation of the GP-19 in Z19, H19 or WH19 terminals. Skip Sections 3.1.2 and 3.1.3 and proceed to Section 3.1.4 for testing and adjustments.

## INSTALLATION

### 3.1.2 GP-19 Installation In Z19-CN, Z19-HN Or H19A Terminals

To complete steps 1-3, installers with little or no technical expertise should refer to the "Notes to Non-Technical Installers" in Section 3.1.

1. Transfer the 40 pin IC from location U430 on the Zenith logic board to location U27 near the top center of the GP-19 logic board. U430 on the Zenith logic board is positioned approximately 4 inches from the top of the logic board and just to the right of center. Make sure the IC is inserted with pin 1 toward the right side of the GP-19 logic board. Note: The serial number appears on the "left" side of the GP-19 logic board, and the metal heatsink is located on the "top" of the board. Pin 1 of the IC is identified in Figure 3.2.



Figure 3.2

2. Transfer the 40 pin IC from location U417 on the Zenith logic board to location U50 near the top left of the GP-19 logic board. U417 is approximately 4 inches to the left of the IC removed in step 1. Make sure the IC is inserted with pin 1 toward the right side of the GP-19 logic board. (If the GP-19 was shipped with an IC in location U50 then omit this step.)
3. 

\*\*\*\*\* IMPORTANT \*\*\*\*\*

Recheck to make sure the ICs transferred to the GP-19 logic board have pin 1 correctly positioned. (See Figure 3.2.) Pin 1 should be on the right end of the IC as viewed from the component side of the GP-19 logic board. The writing on the ICs should appear upside-down.
4. Remove the foam strip protecting the pins of the 40 pin DIP-plug connected to the processor interconnect cable supplied with the GP-19.
5. Insert the 40 pin DIP-plug connector of the processor interconnect cable into the socket on the Zenith logic board that was vacated in step 1 (U430). This plug should be inserted so that the attached cable enters the plug from the top of the terminal. Make sure all 40 pins are inserted into the socket and that no pins bend up underneath the DIP-plug.

## INSTALLATION

6. To prevent the ribbon cable from interfering with the installation of the GP-19 Logic board fold it backwards over the top of the Zenith Logic board.
7. Disconnect the multi-colored video/power cable(s) connected to the upper left corner of the Zenith Logic board.
8. Connect either end of the power interconnect cable supplied with the GP-19 to the connector in the upper left corner of the Zenith Logic board. (Note: this cable is "keyed" and will only connect at the proper position - the right side of P401.)
9. Remove the two 6-32 screws from the black plastic inserts in the metal heatsink at the top of the GP-19 Logic board.
10. Place the GP-19 Logic board just in front of the Zenith Logic board with the component side of the board towards the front of the terminal. The GP-19 Logic board should rest in the two slots at the bottom of the terminal. (Note: It is acceptable for the GP-19 Logic board to contact the back of the CRT connector.)
11. Secure the GP-19 metal heatsink to the upright mounting brackets using the two screws removed in step 9.
12. Fold the 40 conductor processor interconnect cable over the front of the GP-19 board and connect it to connector P2 on the GP-19 Logic board. The connectors should be pressed firmly together. Make sure that the connectors are not "offset" with only one row of pins connected.
13. Connect the power interconnect cable to connector P7 on the GP-19 Logic board. This connector is keyed and will connect only if properly oriented.
14. Connect the multi-colored video/power cable(s) from the terminal (disconnected in step 7) to connector P5 on the GP-19 Logic board. These cables connect side-by-side in exactly the same way they did on the Zenith Logic board. They are fully keyed and will connect in only one orientation.

## NOTE

Z19-HN and H19A terminals: If the video/power cable originally was routed over the top of the Zenith Logic board it may be necessary to re-route the cable around the outside of the left mounting bracket in order to reach connector P6 on the GP-19 Logic board.

## INSTALLATION

This completes the installation of the GP-19 in Z19-HN or H19A terminals. For Z19-CN terminals proceed to Section 3.1.3. For Z19-HN or H19A terminals, skip Section 3.1.3 and proceed to Section 3.1.4 for testing and adjustments.

### 3.1.3 GP-19-CN Power Supply Installation (Z19-CN Only)

Installation of the GP-19 in Z19-CN terminals requires the addition of a power supply module. The power supply is mounted in the right rear corner of the terminal on an existing power supply platform. Installation of this power supply (which is shipped separately from the GP-19 logic board) requires both a screwdriver and a pair of pliers.

1. 

\*\*\*\*\* WARNING \*\*\*\*\*

Make sure the terminal is unplugged from the AC Line before installing the power supply module.
2. Identify each of the following components shipped with the GP-19-CN power supply:
  - \* (1) GP-19-CN power supply circuit card
  - \* (1) Power supply cable
  - \* (5) Self-stripping in-line tap connectors (1 spare)
  - \* (4) #12 X 0.5 inch self-tapping screws
3. Locate the four power supply mounting holes. These are located in the four standoffs on the power supply mounting platform in the right rear corner of the terminal.
4. Pre-thread each of the four power supply mounting holes by driving one of the self-tapping #12 screws part way into each hole and then removing the screw.
5. Mount the power supply board with the diagonal corner of the board located at the rear of the terminal. Secure with the four #12 screws provided.
6. Attach the power supply cable to connector P1 of the power supply. The connector is keyed and will attach only if it is correctly oriented.
7. Using the self-stripping in-line tap connectors (see instructions on the following page), connect the four black wires from P1 to the matching wires entering the terminal's



## INSTALLATION

transformer. (The transformer is located just in front of the newly installed power supply.)

### NOTE

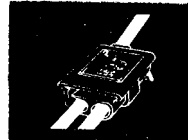
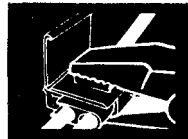
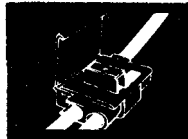
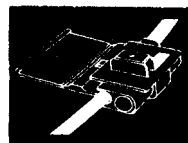
The wires from P1 and the transformer actually have four distinct wire colors. They are:

- \* Black with a red stripe
- \* Black with a green stripe
- \* Black with a yellow stripe
- \* Black with no stripe

It is extremely important to match each wire color from connector P1 with the corresponding wire color from the terminal's transformer.

### Instructions for Using Self-Stripping In-Line Tap Connectors:

- a) Place transformer wire inside run channel. (Do not cut or strip wire.)
- b) Close side cover until latched. Insert wire from P1 (do not strip wire) and check position in inspection port.
- c) Make connection with pliers by driving "U" contact down flush with top of connector.
- d) Close hinged top cover until latched.



8.

### \*\*\*\*\* IMPORTANT \*\*\*\*\*

Recheck each wire from connector P1 on the power supply to ensure it is connected to the wire with the same color stripe on the terminal's transformer.

## INSTALLATION

9. Route the 18 inch cable from the power supply under the neck of the display tube and connect it to P3 near the upper left corner of the GP-19 logic board.

This completes the installation of the GP-19 in Z19-CN terminals. Proceed to Section 3.1.4 for testing and adjustments.

## INSTALLATION

### 3.1.4 Testing And Adjustments

In the event any of the following test operations fail, consult Section 3.1.5 and Chapter 5. The following procedure is recommended to verify that the GP-19 has been correctly installed and is functioning properly:

1. Reconnect the terminal to the AC line.
2. While holding the LEFT SHIFT key down, turn the power switch on. Continue holding the left shift key down until the terminal "ticks" (at which point the SHIFT key should be released). The terminal is now running "Power-up Diagnostics" (see Section 5.2). The four LED indicators in the upper right corner of the GP-19 logic board should all light up and remain lighted for approximately 17 seconds, after which they should all be extinguished and the terminal should function normally. If the LEDs fail to light up or to be extinguished after 17 seconds, or if the terminal's bell is sounded after 17 seconds, proceed to Section 3.1.5.
3. Enter "Set-up mode" by pressing the ESC key while holding down the CTRL key. Verify the presence of text on the screen. Exit "Set-up mode" by again pressing the ESC key while holding down the CTRL key.
4. To test "wide-screen mode" press the "RED" key while holding down the CTRL key, then press ESC while holding down the CTRL key. The text on the screen should appear much narrower. Exit "Set-up mode" by again pressing ESC while holding down the CTRL key.
5. To test "Graphics mode" (Alpha state) press the "WHITE" key while holding down the CTRL key. A blinking block cursor should appear.
6. To test the vector drawing capabilities of the GP-19 press the "WHITE" key while holding down both the CTRL key and a SHIFT key. The cursor should disappear. Key in the following sequence with the terminal off-line (be sure to use lower case letters and include the indicated spaces):  
0 0p 511 0v 511 249v 0 249v 0 0v 511 249v 0 249p 511 0v.  
The terminal should draw a box with diagonals.
7. Return the terminal to 80 column mode by pressing the "BLUE" key while holding down the CTRL key.

## INSTALLATION

A keyboard label designed to facilitate use of the added functions of the GP-19 has been provided. This label is intended to mount just below the CRT on the face of the terminal. It should be positioned so that the leftmost key appearing on the label is above the "ESC" key and the rightmost key appearing on the label is above the "WHITE" key. The surface to which the keyboard label is to be mounted should be free from dust and grease. The label has a pressure sensitive adhesive on the back that becomes very permanent after a short time.

The only adjustment required by the GP-19 is the establishment of the correct width vs height (aspect) ratio. This is necessary if circles are to appear round rather than elliptical. To adjust for the proper aspect ratio do the following:

1. Place the terminal in graphics mode by pressing the "WHITE" key while holding down the CTRL key.
2. Place the terminal in "reverse screen state" by pressing the F4 key while holding down the CTRL key.
3. Measure the width and height of the displayed area. The width should be 1.3 times the height. Correct adjustment usually results in a width of approximately 8 inches and a height of approximately 6 inches. Refer to the section on readjustment in the VIDEO TERMINAL OPERATION MANUAL supplied by Zenith if the terminal requires adjustment of the height or width.

## INSTALLATION

### 3.1.5 In Case Of Difficulty

Each GP-19 is thoroughly tested before shipping - so failure of the GP-19 to function can most probably be attributed to an installation problem. The most common installation faults are:

- \* Incorrectly socketed ICs

Check that the ICs transferred to the GP-19 logic board are inserted correctly with all pins in the sockets. Make sure that pin 1 is correctly positioned (see Figure 3.1 or Figure 3.2). Sometimes a pin will bend up underneath the IC. This can be very difficult to detect except by removing the IC from its socket. If this occurs, straighten the bent pin and re-insert the IC into the socket.

- \* Incorrectly socketed DIP-plug

Make sure all 40 pins of the DIP-plug are inserted into the socket and that no pins bend up underneath the DIP-plug.

- \* Incorrectly socketed cables

It is possible to connect the processor interconnect cable at connector P2 with only one row of pins making contact. Inspect to make sure this has not occurred and that the connectors are pressed completely together.

- \* A few versions of the Z19-CN terminal have a large capacitor mounted at the bottom center of the Zenith logic board. This capacitor can block the GP-19 logic board from resting in the groves at the bottom of the terminal (see step 10 in Section 3.1.2). If this problem exists, contact Northwest Digital Systems for a suitable replacement capacitor.

See Chapter 5 (Troubleshooting) if the GP-19 fails to operate after rechecking the installation procedure.

## INSTALLATION

### 3.2 SWITCH SETTINGS

The GRAPHICS-PLUS logic board includes a group of eight DIP switches labeled SW1 and located near the upper right corner. These switches determine the state of the terminal after power up or reset. The individual switches are designated 1-8. On some boards the switch positions are labeled "on" and "off" rather than "0" and "1". In this case, "on"= "0" and "off"="1".

Switches 1 and 2 determine what trailer code character or characters are transmitted at the end of a Tektronix mode status/position message. The possibilities are as follows:

| <u>SWITCH 1</u> | <u>SWITCH 2</u> | <u>CODE(S) TRANSMITTED</u> |
|-----------------|-----------------|----------------------------|
| 0 (on)          | 0 (on)          | none                       |
| 0 (on)          | 1 (off)         | *CR                        |
| 1 (off)         | 0 (on)          | none                       |
| 1 (off)         | 1 (off)         | CR EOT                     |

(\*) Indicates the factory setting.

## INSTALLATION

Switches 3-8 have the following effects:

### SWITCH    EFFECT

- 3    0 (on) = Normal Z19 SHIFT key  
      \*1 (off)= The SHIFT key will invert the effect of the CAPS LOCK key as described in paragraph 4.12.4.
- 4    \*0 (on) = The terminal transmission rate will be limited only by the baud rate.  
      1 (off)= The terminal will limit its transmission rate to the host to one character per 16.7 milliseconds. On some computer systems this will prevent buffer overflows when the terminal transmits long character sequences. See paragraph 4.12.8.
- 5    0 (on) = VT100 character set selected (see paragraph 4.12.11).  
      \*1 (off)= Zenith character set selected.
- 6    \*0 (on) = When the terminal is in ANSI escape mode it will be in the Z19 ANSI configuration.  
      1 (off)= When the terminal is in ANSI escape mode it will be in the EDT compatible configuration (see paragraphs 4.12.10 and 4.13.1.1).
- 7    \*0 (on) = When the terminal is in Tektronix mode the cursor will wrap at column 74.  
      1 (off)= When the terminal is in Tektronix mode the cursor will wrap at column 73 (see paragraph 4.6.1.2).
- 8    \*0 (on) = Screen timeout enabled. The screen will automatically be blanked if no characters are received or typed for 10 minutes (see paragraph 4.12.6).  
      1 (off)= Screen timeout disabled.

(\*) Indicates the factory setting.

All of the switches on the Z-19 terminal logic board retain their original functions except for S402 section 7. This switch, which formerly selected 50 HZ or 60 HZ vertical sync rate, now enables/disables the automatic entry of Holdscreen operation when SHIFT-SCROLL is depressed (see paragraph 4.12.1). If this switch is set to "1" the feature is enabled.

## CHAPTER 4

### OPERATION

#### 4.1 OVERVIEW

GRAPHICS-PLUS has been designed in such a way as to preserve the original functions of the Z19 terminal while adding additional features. The OPERATION chapter of this manual is intended to describe these additional features, not the native functions of the Z19 which are described in the VIDEO TERMINAL OPERATION MANUAL supplied by Zenith.

A Z19 equipped with GRAPHICS-PLUS has eight distinct modes of operation. These include four different text display modes, two graphics modes, a Transparent mode that allows display of all received information including control characters, and a Set-up mode for configuring the terminal. A set of local function keys is available which allows the operator to change to any mode using a single key stroke.

GRAPHICS-PLUS also adds off-screen display memory to the Z19. Off-screen memory means that when text scrolls off the screen it is not "gone forever", it is simply scrolled into a portion of display memory not presently being displayed. GRAPHICS-PLUS provides a total of 16,240 characters of display memory, about 203 lines when operating in an 80 column display mode. Thus information is not truly "gone forever" until it scrolls out of the un-displayed portion of memory as well. A set of local function keys allows the user to control what portion of the display memory is currently visible on the screen.

Much of the information in this chapter is more detailed than will be required by the average user. If the user simply wishes to employ the terminal in conjunction with existing Tektronix compatible programs, then there is very little special knowledge needed. The terminal automatically enters Tektronix mode upon receipt of a GS control character or the character sequence (ESC FF), so most Tektronix compatible programs will function correctly with no special operator action. The user need only know that CTRL-BLUE will return the terminal to the normal 80 column text mode of operation. If, however, the user is developing new graphics programs, then the information on extensions



## OPERATION

to the Tektronix protocol supported by GRAPHICS-PLUS and found in Section 4.6 may prove useful.

Users employing the terminal for routine applications can gain an understanding of the major features by reading the first five sections of this chapter (4.1 thru 4.5). These sections explain local function keys, the four text formats, Set-up mode, and off-screen memory. Sections 4.6 thru 4.8 contain a detailed description of the graphics modes. Sections 4.9 thru 4.14 describe other features which will be of interest to the more sophisticated user. In particular, Section 4.12 describes a number of additional features such as the VT100 style scroll key, the "smart" shift key, and the screen timeout.

Many of the features, such as the programmable function keys and the programmable tab stops, can be controlled both locally from Set-up mode (see Section 4.3) and remotely using escape sequences. Information on the use of escape sequences to control particular terminal features can be found in the manual sections describing those features. Additional information on escape sequences can be found in Section 4.13 and Appendix A.

### 4.2 LOCAL FUNCTION KEYS

The nine keys on the Z19 keyboard labelled in yellow F1-F5, BLUE, RED, WHITE and ESC now have additional functions. A keyboard decal is provided to label these keys with their new functions. Although the keys still retain their original functions, they take on new meaning when used in conjunction with the CTRL and SHIFT keys. Figure 4.1 shows the new functions of the keys. These keys are called "local function keys" because when used with either CTRL or CTRL and SHIFT they do not transmit a character or sequence of characters to the host computer; instead they affect only the mode of operation of the terminal and the current display. The local function keys allow the user to conveniently change terminal modes, change the display format, and manipulate off-screen memory.

FIGURE 4.1

#### Local Function Keys



## OPERATION

The following local functions are provided:

1. CTRL-F1: Scroll the display backward one line.
2. CTRL-SHIFT-F1: Scroll the display backward one page.
3. CTRL-F2: Scroll the display forward one line.
4. CTRL-SHIFT-F2: Scroll the display forward one page.
5. CTRL-F3: Scroll to the end of display memory.
6. CTRL-SHIFT-F3: Scroll to the beginning of display memory.

Further information on the use of the local functions to control off-screen memory can be found in Section 4.5.

7. CTRL-F4: Set/clear (toggles) reverse screen state. See Section 4.11 for details.
8. CTRL-SHIFT-F4: Activate the optional hardcopy accessory.
9. CTRL-F5: Clear the display memory from the current cursor position to the end.
10. CTRL-SHIFT-F5: Clear all of display memory.
11. CTRL-ERASE: Erase to end of page; clears the portion of the display memory extending from the cursor position to the end of the screen.
12. CTRL-SHIFT-ERASE: Erase page; clears the portion of the display memory currently displayed on the screen.
13. CTRL-BLUE: Select 80 column by 24 line display format.
14. CTRL-SHIFT-BLUE: Select 80 column by 49 line display format.
15. CTRL-RED: Select 132 column by 24 line display format.
16. CTRL-SHIFT-RED: Select 132 column by 49 line display format.

Further information on display formats can be found in Section 4.4.

## OPERATION

17. CTRL-WHITE: Change to Tektronix mode Alpha state.  
See Section 4.6 for details.
18. CTRL-SHIFT-WHITE: Change to Easy-Entry Graphics mode.  
See Section 4.7 for details.
19. CTRL-ESC: Change to/exit from (toggles) Set-up mode.  
See Section 4.3 for details.
20. CTRL-SHIFT-ESC: Change to/exit from (toggles) Transparent mode. See Section 4.9 for details.

In addition to these local functions, when the terminal is in either of the two graphics modes the ERASE key will perform a screen erase without the use of SHIFT or CTRL. If the ERASE key is used while in Easy-Entry Graphics mode the terminal will be in Tektronix mode Alpha state when the erase operation is complete.

### 4.3 SET-UP MODE

Set-up mode provides a convenient way for the user to control many of the programmable aspects of the terminal that are not under the direct control of local function keys. Set-up mode is fully menu driven and requires very little prior knowledge to use. Set-up mode can be entered from any other terminal mode by using the local function key CTRL-ESC. When in Set-up mode the local function key CTRL-ESC can be used to exit Set-up mode. If Set-up mode is entered from a text mode, when Set-up mode is exited the terminal will return to the previous mode and the prior screen contents will be restored. If Set-up mode is entered from a graphics mode, then exiting Set-up mode will return the terminal to the last used text mode, rather than the graphics mode, and the graphics display will be lost. Entering Set-up mode automatically enables the keyboard and the screen display if either of these was in a disabled state. Exiting Set-up mode re-enables character transmission from the terminal output buffer if transmission had been halted by receipt of a DC3 (CTRL-S, "XOFF") control character from the host computer (see Section 4.12.13). Characters that arrive from the host while the terminal is in Set-up mode are not lost, instead they are stored in the terminal input buffer (see Section 4.14).

When the terminal enters Set-up mode the screen displays a menu of the programmable parameters and their current values. The current values are displayed in reverse video. A blinking arrow points to the parameter that is currently selected for alteration. The selected parameter is altered by pressing the space bar; each time the bar is pressed the displayed value of the parameter will change to the next possible alternative value. For most parameters there are only two possible values: ENABLED and DISABLED. A few parameters, such as baud

## OPERATION

rate, have more. The arrow is moved to select a different parameter by using the directional keys on the numeric keypad: the arrow moves to the next available parameter in the direction indicated by the label on the keytop. The HOME key moves the arrow back to the first parameter in the menu.

The last parameter on the menu, TAB STOPS, has two possible values: STANDARD and USER'S. When the value is set to STANDARD the positions of the default Z19 tab stops are displayed beneath a ruler at the bottom of the screen. If the parameter value is changed to USER'S then the display of the default tab stops is replaced by a display of the user's programmed tab stop positions. While user programmed tab stops are enabled the arrow can be moved to a position beneath the ruler, whereupon it becomes a blinking square. The blinking square is moved along the ruler by the horizontal direction keys on the numeric keypad, and tab stops can be set or cleared by striking the space bar or the TAB key. Multiple tab stops at regular intervals can be set by moving the cursor to the desired position for the first stop and striking REPEAT-TAB. This sets a tab stop at the position of the cursor and at all integer multiples of that position. For instance, if the cursor was at column 9, the result would be stops at columns 9, 17, 25, 33, 41, 49, 57, 65, 73, 81, 89, 97, 105, 113, 121, and 129. In addition, there is always an implicit tab stop at the last column (either 80 or 132, depending on the display format). All tab stops can be cleared at once by striking the DL key on the numeric keypad.

The first parameter on the menu, MENU, indicates what menu is currently being displayed. If the value of this parameter is changed by striking the space key a new Set-up menu will appear. Unless an option board has been installed, there is only one alternative menu: PROGRAMMABLE FUNCTION KEYS.

While in the PROGRAMMABLE FUNCTION KEYS menu the arrow can only be moved between two parameters: the first controls the menu, and the second controls whether the selected function key is to be PROGRAMMED, DISPLAYED, or RESET. The desired function key is selected simply by striking it (with SHIFT, if the shifted function of the key is to be selected). If the key is to be DISPLAYED then the current programmed string for the key will be displayed at the bottom of the screen. Any control characters in the string are displayed in reverse video. If the key is to be RESET then the programmed string will be reset to the default Z19 escape sequence for the key (depending on the current escape mode of the terminal, either an ANSI or a ZENITH sequence will be used) and that will be displayed. If the key is to be PROGRAMMED then the current value is displayed and the user can enter new characters. Any ASCII character is valid for inclusion in the string, and the string can be edited by use of the horizontal direction keys on the numeric keypad. A blinking square appears as a cursor to indicate the position in the string where the next character typed will be entered. The "." key on the numeric keypad will move the cursor to the end of the string and the

## OPERATION

DL key will delete the string. The ENTER key will terminate the programming operation. Only those characters that are to the left of the cursor at the time ENTER is struck will be included in the string. Once programming of a function key has commenced it can be aborted without effect by simply striking the function key again. Programming any function key while in Set-up mode automatically sets the Programmable Keys Active state of the terminal.

Function key strings can be edited using the IC and DC keys on the numeric keypad. IC will insert a space at the current cursor position and shift the remainder of the text (including the character that was under the cursor) one position to the right. Thus characters can be added to a string by first inserting spaces and then overwriting the spaces with new characters. DC will delete the character at the current cursor position and shift all text to the right of the cursor one character to the left.

It is possible to directly include the default escape sequences generated by the function keys F1-F5 and the colored keys (BLUE, RED, and WHITE), and those generated by the keypad, as part of the character string. To do this, simply strike the desired key while holding down CTRL, and the generated sequences will appear as part of the string. Note that for this to work correctly, the terminal must already be set to the necessary state to generate the desired sequences (i.e., ANSI or ZENITH escape mode, alternate or normal keypad, etc.). This technique can be used to program function keys as screen editing macros for use with some editors.

While the terminal is in the Set-up mode PROGRAMMABLE FUNCTION KEYS menu, striking CTRL-V causes the version number of the GRAPHICS-PLUS firmware to be displayed on the bottom line of the display.

### 4.4 TEXT MODE FORMATS

The terminal has four different text display format modes. The text display format mode of the terminal can be changed either by the action of a local function key or by receipt of an appropriate escape sequence (see Appendix A for escape sequences). The local function keys that can be used to enter each of the text display format modes are as follows:

- |    |                         |                 |
|----|-------------------------|-----------------|
| 1. | 80 columns by 24 lines  | CTRL-BLUE       |
| 2. | 80 columns by 49 lines  | CTRL-SHIFT-BLUE |
| 3. | 132 columns by 24 lines | CTRL-RED        |
| 4. | 132 columns by 49 lines | CTRL-SHIFT-RED  |

Changing between a 80 column mode and a 132 column mode (or vice versa) automatically clears the display memory. Changing between a 24 line mode and a 49 line mode (or vice versa) does not clear the screen; instead, the most recent 24 (or 49) lines of text in memory are

## OPERATION

displayed. In all modes the "25th line" feature of the Z19 is still available, however in the 49 line modes it is the "50th line" (see paragraph 4.12.12).

The 49 line modes employ an interlaced display which results in a perceptible flickering of the screen. This is a normal and unavoidable feature of operating in these modes. Operating the terminal at lower brightness considerably reduces the amount of perceived flicker.

### 4.5 OFF-SCREEN MEMORY

The terminal has 16,384 characters of display memory, enough for 203 lines of text in 80 column mode or 123 lines of text in 132 column mode. The screen is a window into this memory that can only display a single page of the contents at any one time; a page being either 24 or 49 lines depending on the vertical display format. By default, the screen window is automatically positioned to display the text most recently entered into memory. However, local function keys or escape sequences can be used to position the window so as to display any page of text currently in display memory.

After the terminal is reset or the display memory is cleared, the screen window is positioned to the beginning of display memory and the cursor is at the upper left corner of the screen. As each new line of text is placed in the display memory the cursor moves down one line on the screen. When the screen window becomes full the next line placed in memory causes the window to move down one line, so that the new line will be displayed at the bottom of the screen. This is referred to as "scrolling forward", since moving the window down is accomplished by moving it forward towards the end of memory. After this scrolling operation, the top line of the page displayed in the screen window is line two of the display memory; line one is still in the display memory but is no longer visible in the window. Every successive new line is handled in this manner until the display memory becomes full. Once display memory is full, the oldest line in memory must be discarded to make space available each time a new line arrives. This is done by scrolling the display memory, i.e., each line in memory is moved backward one line (toward the beginning) and the oldest line is discarded. The screen window remains positioned to display the last page in memory, that is, the most recent 24 or 49 lines.

The user can use local function keys or escape sequences to position the screen window in off-screen memory. This is most commonly done to bring back into view text that has already scrolled off the top of the screen. These functions work as described in the following paragraphs.

## OPERATION

### 4.5.1 Scrolling Backward

Scrolling text backward moves the screen window back towards the beginning of memory, bringing older lines onto the screen. The local function key CTRL-F1 scrolls text backward one line, while CTRL-SHIFT-F1 scrolls text backward one page. The equivalent global escape sequences are:

ESC : Pn B        Scroll backward Pn line(s)  
ESC : B            Scroll backward a page

It is also possible to move the screen window directly to the beginning of memory by using the local function key CTRL-SHIFT-F3. The equivalent function can be accomplished with the global escape sequence (ESC : Pn B) by specifying a Pn greater than or equal to the maximum number of lines in display memory (ESC : 203 B), or by using the absolute positioning global escape sequence (ESC : 0 a).

It is impossible to scroll backward past the beginning of memory. Once the screen window has reached the point where the first line in memory is displayed at the top of the screen, any further scroll backward commands will be ignored.

### 4.5.2 Scrolling Forward

Scrolling text forward moves the screen window forward towards the end of memory, bringing newer lines onto the screen. The local function key CTRL-F2 scrolls text forward one line, while CTRL-SHIFT-F2 scrolls text forward one page. The equivalent global escape sequences are:

ESC : Pn F        Scroll forward Pn line(s)  
ESC : F            Scroll forward a page

The local function key CTRL-F3 moves the screen window forward directly to the end of memory. This is usually the quickest and most convenient way to return to "where you left off" after having scrolled back in memory to examine old lines previously scrolled off the screen. Moving the screen window directly forward to the end of memory can be accomplished by either of two functionally equivalent global escape sequences:

ESC : a  
ESC : 203 F

It is not possible to scroll forward past the logical end of memory. The logical end of memory is the last line in memory that has actually been used. If the memory has not been filled since the terminal was last reset or the display memory was cleared, then the logical end of memory will be at some point before the physical end of memory. Once

## OPERATION

the screen window has reached the point where the bottom line displayed on the screen corresponds to the logical end of memory, all further scroll forward commands will be ignored.

### 4.5.3 Scrolling Speed

Scroll forward and scroll backward operations are performed "slowly" or "quickly" depending on whether or not Slow Page Scroll is enabled. If Slow Page Scroll is enabled (it is by default) , scroll forward and scroll backward operations are performed at a rate of about one page per second. If Slow Page Scroll is disabled, scroll forward and scroll backward operations are performed as instantaneous discrete jumps. Scrolling "slowly" is more time consuming but provides a sense of visual continuity that is lost when scrolling is done in discrete jumps. Slow Page Scrolling can be disabled by the global escape sequence (ESC : 3 l) and enabled by (ESC : 3 h). Operations which move the screen window directly to the beginning or end of memory are not affected by the state of Slow Page Scroll; they are always performed as a discrete jump.

### 4.5.4 Inserting Lines In Display Memory

Normally, new lines of text are always inserted at the logical end of display memory. If, however, the screen window has been moved so that the cursor position no longer corresponds to the logical end of memory, this is no longer true. For example, if the screen window has been moved so that the bottom line is displaying line 100 of memory, and the cursor is at the bottom of the screen, then the next new line will be inserted at line 100 in memory. The new line will overwrite the existing contents of line 100 in memory. If the new line is less than a full screen width long, then the non-overwritten portion of the old line will remain on the screen. The next line feed operation (or automatic wrap) will cause the screen window to move forward one line, so that line 101 of memory is now on the bottom of the screen. As is always the case with a line feed operation when the cursor is at the bottom of the screen, the next line of memory is cleared before it is scrolled onto the screen. If the cursor is not at the bottom of the screen, then line feed operations move the cursor down but do not clear a line of memory.

There is one additional escape sequence available for positioning the screen window in display memory. This sequence (ESC : Pn a) positions the screen window so that the home line (top line on the screen) is the absolute line in display memory specified by Pn. The first line in display memory has the absolute line number one. The last line has the absolute line number 203 (80 column mode) or 123 (132 column mode). Since it is not possible to position the screen window past the logical end of memory, and the screen window contains either 24 or 49 lines, the maximum legal screen home line positions are as follows:



## OPERATION

| <u>Display Format</u>  | <u>Max. Legal Address</u> |
|------------------------|---------------------------|
| 80 column by 24 lines  | 180                       |
| 132 column by 24 lines | 100                       |
| 80 column by 49 lines  | 155                       |
| 132 column by 49 lines | 75                        |

### 4.5.5 Clearing Display Memory

Four local function keys are available for clearing display memory. CTRL-F5 clears display memory from the current cursor position to the end of memory. The equivalent global escape sequence is:

ESC : 0 E

The local function key CTRL-SHIFT-F5 clears all of display memory. The equivalent is:

ESC : E

The local function key CTRL-ERASE clears the portion of display memory extending from the cursor position to the end of the screen window. There is no global escape sequence to perform this function although there are native Z19 ANSI and ZENITH escape sequences that do.

The local function key CTRL-SHIFT-ERASE clears the portion of display memory currently in the screen window. The equivalent global escape sequence is:

ESC : 1 P

The affect that this operation has on the position of the cursor is determined by whether EDT compatibility is set (see paragraph 4.13.1.1) and whether the terminal is in Absolute or Relative Origin mode (see paragraph 4.12.9.1).

And finally, the global escape sequence (ESC : 0 P) clears the portion of display memory corresponding to the currently defined scrolling region (see Section 4.12.9 for an explanation of the definable scrolling region). The cursor is placed at the start of the defined scrolling region.

The escape sequence shown in the previous two examples, which has the general form (ESC : Pn P), is also recognized in Tektronix mode and has the same affect as the Tektronix mode erase screen escape sequence (ESC FF). In this case the value of Pn is ignored.

## OPERATION

### 4.6 TEKTRONIX MODE

When operating in Tektronix mode the terminal emulates all the features of a Tektronix model 4010 computer display terminal. The Tektronix 4010 is a storage-tube based terminal that displays upper-case alphanumeric text in a 74 character by 35 line non-scrolling format and displays graphical information represented as a series of vectors drawn in a 1024 point wide by 780 point high graphics space. Tektronix mode can be entered directly by use of the CTRL-WHITE local function key. The terminal automatically enters Tektronix mode if it receives a GS control character or the escape sequence (ESC FF) while in any text mode.

GRAPHICS-PLUS emulates the 4010 by using a refreshed display, rather than a storage tube, presenting an image composed of 128,000 dots arranged in a 512 dot wide by 250 dot high grid. Each of these dots represents a unique bit in the terminal display memory and can be turned on (or off) independently of every other dot in the display. The internal Z80 microprocessor displays characters and vectors by setting the necessary bits in display memory to create the correct image.

The screen of a 4010 terminal has an addressable display area 1024 "tekpoints" wide by 780 "tekpoints" high (the maximum legal vertical address is actually 1023, but only 780 tekpoints fall in the display area). The lower left corner corresponds to address 0X, 0Y and the upper right corner is address 1023X, 779Y. Vectors to be drawn are encoded by specifying their end points using this addressing scheme. Furthermore, the position of the display cursor is defined by specifying the address of the lower left corner of the cursor. The cursor is a blinking solid square 22 tekpoints high by 14 tekpoints wide.

Since the 512 x 250 dot resolution of GRAPHICS-PLUS is less than that inherent in the Tektronix 4010 addressing scheme, a formula is employed to translate between the two coordinate systems. The Z80 microprocessor converts display addresses specified in "tekpoints" to positions on the GRAPHICS-PLUS display using the following relations:  $X' = X/2$  and  $Y' = (Y*41+18)/128$ , where X and Y are the positions specified in "tekpoints" and X' and Y' are GRAPHICS-PLUS screen positions.

When operating in Tektronix mode the terminal can be in any one of three possible states: Alpha state, Graph state, or GIN state. In Alpha state the terminal displays received information as alphanumeric characters, in Graph state it interprets received information as vector end point coordinates and displays the resulting vectors, and in GIN (Graphics INput) state it returns information about the state of the terminal to the host computer. Although there are control characters and escape sequences defined for entering the various Tektronix mode states, Tektronix mode Alpha state can be entered directly by use of the CTRL-WHITE key.

## OPERATION

When in Tektronix mode the ERASE key is a local function key that clears the screen, sets Alpha state, and returns the Alpha state cursor to the upper left hand corner of the screen. The function of this key is identical to the "Page" key on a 4010 terminal. The local function key CTRL-SHIFT-F5 also clears the screen but does not set Alpha state or affect the cursor position. The local function key CTRL-F4 reverses the video level so that characters and graphics are displayed as black on a white background.

When the terminal is in Tektronix mode the keys on the numeric keypad portion of the keyboard can only be used for GIN state cursor positioning (see Section 4.6.3). The keys on the numeric keypad can not be used for data entry while the terminal is in Tektronix mode. The rest of the keyboard functions normally.

### 4.6.1 Alpha State Operation

When in Alpha state of Tektronix mode, the terminal displays received information as upper-case characters using a 7x7 dot matrix character cell. The display format is 73 characters wide by 35 lines high. Due to interactions with Graph state to be explained later, characters can in fact be placed on the screen at arbitrary locations that do not correspond to positions in this grid. The display does not scroll when operating in Alpha state of Tektronix mode, instead the display is treated as two alternating "columns". Assuming that the cursor starts out in the "home" position (OX, 767Y), the first 35 lines of text received will be displayed on successive lines each starting at the left edge of the screen. The left edge is referred to as "Margin 0". The 36th line will be displayed on the top line of the screen with the first character of the line in column 37. This is referred to as "Margin 1". The next 34 lines will be displayed on successive lines, each beginning at Margin 1, until the bottom of the screen is reached. At this point the terminal will begin displaying text at Margin 0, and so on. Note that when a character "overwrites" a character already on the screen the effect is just as it would be on a storage tube display: what you see is the combined image of the two characters. In order to maintain a legible display, the user can clear the screen with the ERASE key. The blinking block cursor which is displayed in Alpha state can be disabled (turned off), or re-enabled, with the same escape sequences that control the text mode cursor (see Section A.16).

If a line extends beyond the the 74th character position of a line, an automatic RETURN/LINE-FEED operation is performed by the terminal (see paragraph 4.6.1.2). There is also a switchable option that will force the terminal to perform a RETURN every time a LF is received (see paragraph 4.6.5, "LF"), and an option to perform a LINE-FEED upon receipt of a CR (see paragraph 4.6.5, "CR") .

## OPERATION

4.6.1.1 Interaction With Graph State - If the terminal returns to Alpha state from Graph state because it received a US control character, the alpha cursor will appear at a point on the screen such that the lower left hand corner of the cursor corresponds to the last vector end point in Graph state. If the terminal returns to Alpha state from Graph state because it received a carriage return, the alpha cursor will appear at the margin (see Paragraph 4.6.5, "CR") with its vertical position the same as if a US control character had been received. If switch 402 section 3 on the Z19 logic board is set to "1" (see paragraph 4.6.5, "CR"), the vertical position after a carriage return is modified by the resulting automatic line feed operation. In any case, subsequent characters will be displayed at positions that are relative to this new cursor position. The user can utilize this effect to position character strings on the screen at arbitrary positions that cannot be reached if the cursor starts out at the home position.

4.6.1.2 Line Length - Due to the interaction of the horizontal screen format and the character cell size the terminal is only capable of displaying 73 complete characters on a line, not 74 as is the case with a Tektronix 4010. In fact only a fraction of the 74th character will be displayed. If the cursor started out at a margin, this fraction will correspond to only the first vertical column of the character cell, which contains no displayed dots. If the cursor started the line at an arbitrary position, as explained in the previous paragraph, the terminal will display whatever fraction of the last character that will fit on the screen. If the user requires that characters in the 74th column be displayed rather than lost, setting switch SW1 section 7 to the "1" position (see Section 3.2) causes the the cursor to wrap after 73 columns and display the 74th character as the first character on the next line. If this switch is set to "0" the cursor will not wrap until after the 74th column. Note that this switch only affects the terminal in Tektronix mode.

4.6.1.3 Status/position Readback - When operating in Tektronix mode Alpha state, the terminal will respond to receipt of the escape sequence (ESC ENQ) by transmitting a character sequence indicating the terminal status and the position of the lower left corner of the cursor. The format of this sequence is as follows:

1. The Status character, in the form P01H01M1. If bit H is clear it indicates that the optional hardcopy printer is on line. If bit M is set it indicates Margin 1 rather than Margin 0 is set.
2. The most significant five bits of the X address, in the form P01XXXXX.

## OPERATION

3. The least significant five bits of the X address, in the form P01XXXXX.
4. The most significant five bits of the Y address, in the form P01YYYYY.
5. The least significant five bits of the Y address, in the form P01YYYYY.
6. A CR character. Transmission of this character is optionally enabled by switch SW1 sections 1 and 2 (see Section 3.2).
7. A EOT character. Transmission of this character is optionally enabled by switch SW1 sections 1 and 2 (see Section 3.2).

P signifies the parity bit.

Once transmission of the message has started, the terminal will ignore all received characters until one of two events occur:

1. One of the following characters is received: BEL, BS, CR, HT, LF, US, GS, or VT.
2. One of the following character sequences is received: (ESC ETB) or (ESC FF) or (ESC X) or any valid sequence beginning (ESC :).

The practical effect of this is usually to ignore any characters echoed by the host computer until such time as the CR is echoed, which returns the terminal to normal Tektronix mode Alpha state operation. This means that the status/position readback can be invoked by a host computer operating in full duplex without any necessity for suspension of character echoing by the host.

### 4.6.2 Graph State Operation

The terminal enters Tektronix mode Graph state whenever it receives a GS control character (unless, of course, the GS was received while in Transparent mode). When in Tektronix mode Graph state the terminal interprets received information as vector end points and displays the resulting vectors. When the terminal receives information completely specifying a vector end point, it draws a vector starting where the last vector ended and continuing to the newly specified end point. Vectors can be of five different types: light, dark, black, area fill, and area erase. A light vector is a visible line on the display connecting two end points; the line is produced by turning on the necessary dots in the display memory. All displayed graphical images are composed of

## OPERATION

light vectors. Drawing a dark vector produces no visible effect on the screen, it simply updates the "last vector end point" position to the new value. Dark vectors are used to separate light vectors that do not share a common end point. To draw two lines on the screen that are not connected, it is necessary to first draw a light vector (one line), then a dark vector (to move to the start of the second line), and then a second light vector (the second line). A black vector is, in effect, a selective erase; all dots in the path of a black vector are turned off. Thus a line on the screen previously produced by a light vector can be erased by simply "redrawing" it as a black vector. Area fill and area erase vectors are unique in that they operate on rectangular areas of the screen rather than line segments. They are fully explained in paragraph 4.6.2.3. Black, area fill, and area erase vectors are extensions to the Tektronix protocol; they are not recognized by Tektronix terminals.

4.6.2.1 Specifying Vector End Points - The protocol for specifying a vector end point uses a maximum of four characters to specify a new end point. However, if the new vector end point is physically close to the last vector end point, it may require as little as one character to specify it. A complete vector end point specification consists of four characters that are interpreted as follows:

<HI Y> <LO Y> <HI X> <LO X>

These characters are decoded by the terminal to produce two ten bit numbers; one to specify the X position of the vector end point and one to specify the Y position. Since an unsigned ten bit number has a maximum value of 1023, both the X and Y portions of the end point address have possible values in the range of 0-1023. The decoding scheme extracts only five bits of address information from each character. Thus it takes a pair of characters to specify a position on one axis, and four characters to completely specify an end point address. The character that encodes the low order five bits of the position on an axis is referred to as the LO byte (LO X or LO Y), and the character that encodes the most significant five bits is known as the HI byte (HI X or HI Y).

The five bits in a character that are extracted to become part of an end point address are simply the least significant five bits of that character. Of the three remaining bits in the character, the most significant (parity) has no meaning in this protocol. The other two, bits 6 and 5 (numbering the bits with the least significant bit as 0), are used to specify what part of the end point address the character represents; i.e., HI Y, LO Y, HI X, or LO X. The procedure for forming a complete four byte sequence for transmission to the terminal is as follows:

## OPERATION

1. Take the five most significant bits of the desired Y address and affix the prefix bits 01 to create a character of the form:

P01YYYYY (HI Y).

2. Take the five least significant bits of the desired Y address and affix the prefix bits 11 to create a character of the form:

P11YYYYY (LO Y).

3. Take the five most significant bits of the desired X address and affix the prefix bits 01 to create a character of the form:

P01XXXXX (HI X).

4. Take the five least significant bits of the desired X address and affix the prefix bits 10 to create a character of the form:

P10XXXXX (LO X).

"P" designates the parity bit of the character.

As can be seen, the HI Y and HI X bytes carry the same prefix bits; thus the terminal can distinguish between them only by their position in the sequence. The rule for distinguishing them is simple: if a LO Y character has already been received in the current sequence then any character received with a 01 prefix is considered a HI X character, otherwise it is regarded as a HI Y character.

The LO X character terminates a vector end point sequence and causes the terminal to draw a vector ending at the new end point. The simplest possible vector end point sequence thus consists of only one character: a LO X character. Such a sequence specifies a new vector end point that differs from the last vector end point only in the least significant five bits of the X address; it specifies a short horizontal vector. Whenever a sequence terminated by a LO X character is less than four characters long, the terminal regards the unspecified portions of the new end point address as being the same as the corresponding portions of the last end point address. This frequently means that the host does not need to send four characters to specify a vector. If some portion of the address has not changed, it may not need specifying. Table 4.1 shows the sequences that must be transmitted for each of the possible circumstances.

OPERATION

TABLE 4.1

Characters Required When Transmitting Vector Endpoint Addresses

| Address fields |      |      |      | Characters that must |      |      |      |
|----------------|------|------|------|----------------------|------|------|------|
| Changed        |      |      |      | be Sent              |      |      |      |
| Hi Y           | Lo Y | Hi X | Lo X | Hi Y                 | Lo Y | Hi X | Lo X |
|                |      |      | *    |                      |      |      | *    |
|                |      | *    |      |                      | *    | *    | *    |
|                | *    |      |      |                      | *    |      | *    |
| *              |      |      |      | *                    |      |      | *    |
|                |      | *    | *    |                      | *    | *    | *    |
|                | *    |      | *    |                      | *    |      | *    |
| *              |      |      | *    | *                    |      |      | *    |
|                | *    | *    |      |                      | *    | *    | *    |
| *              |      | *    |      | *                    | *    | *    | *    |
| *              | *    |      | *    | *                    | *    |      | *    |
| *              | *    | *    |      | *                    | *    | *    | *    |
| *              | *    | *    | *    | *                    | *    | *    | *    |

4.6.2.2 Specifying The Vector Type - What kind of vector is drawn (light, dark, black, area fill, or area erase) is determined by what preceded the character sequence that specified the new vector end point. If the sequence was preceded by a GS control character, then the vector drawn will be dark. Since the GS control character is also used to cause the terminal to enter Tektronix mode Graph state, it follows that



## OPERATION

the first vector drawn after entering Graph state will always be dark. If a vector end point sequence is preceded by a NAK control character the vector will be drawn black; that is, it will erase all dots in its path. If the preceding character is a FS control character an area fill vector results, and if it is a RS control character an area erase vector results. If the vector end point specification is not preceded by one of these four characters (GS, NAK, FS, or RS), the vector will be drawn light.

4.6.2.3 Area Fill And Erase Vectors - Area fill and erase vectors are radically different from the three other types of vectors; instead of operating along a line segment they operate within a rectangular area. Rather than regarding the last vector end point and the new vector end point as specifying the ends of a line segment, they are taken to specify two diagonally opposed corners of a rectangle. The rectangle uniquely defined by these two corners is either completely filled (all dots within the area are turned on) or erased (all dots within the area turned off). Using an area fill or erase vector, rather than a long sequence of light or black vectors, simplifies the computer program and greatly reduces the number of transmitted bytes necessary to accomplish some screen operations. The amount of time required for the terminal to complete an area fill or erase operation can be estimated using the formula found in Appendix B.

4.6.2.4 Status/position Readback - When operating in Tektronix mode Graph state, the terminal will respond to receipt of the character sequence (ESC ENQ) by transmitting a character sequence indicating the terminal status and the position of the current vector end point. The characters in this sequence encode the information as follows:

1. The Status character, in the form P01H10M1. If bit H is clear it indicates that the optional hardcopy printer is on line. If bit M is set it indicates Margin 1 rather than Margin 0 is set.
2. The most significant five bits of the X address, in the form P01XXXXX.
3. The least significant five bits of the X address, in the form P01XXXXX.
4. The most significant five bits of the Y address, in the form P01YYYYY.
5. The least significant five bits of the Y address, in the form P01YYYYY.

## OPERATION

6. A CR character. Transmission of this character is optionally enabled by switch SW1 sections 1 and 2 (see Section 3.2).
7. A EOT character. Transmission of this character is optionally enabled by switch SW1 sections 1 and 2 (see Section 3.2).

P signifies the parity bit.

The terminal will respond to characters received while that message is being transmitted. Note that this differs from the behavior in Alpha state. If the user wishes the state of the terminal to be undisturbed by the status/readback operation it is essential that the host computer does not echo any of the message characters.

### 4.6.3 Graphics Input (GIN) State Operation

The terminal enters GIN state upon receiving the character sequence (ESC SUB) while in Tektronix mode. When the terminal enters GIN state a cross-hair cursor, approximately 0.5 inch by 0.5 inch in size, will appear on the screen. The GIN cursor appears at the same point on the screen that it occupied when the terminal was last in Tektronix mode GIN state, unless this position has been modified by the escape sequence (ESC X) (see paragraph 4.6.3.1).

The GIN state cursor can be positioned on the screen using the nine keys in the top three rows of the keyboard numeric keypad. Striking the center key in this group of nine (labelled 5 and HOME) moves the cursor to the center of the screen. Striking any of the four keys which are labelled with directional arrows moves the cursor in the direction indicated by the arrow. The remaining four keys (1, 3, 7, and 9) each move the cursor in a diagonal direction; for example, the key labelled 7 moves the cursor up and to the left. Any of the eight directional keys, when struck alone, moves the cursor one dot in the indicated direction. However, if a directional key is struck in conjunction with the left hand SHIFT key, the cursor will move approximately half its width in the appropriate direction. By employing the REPEAT key in combination with the above mentioned keystrokes the user can quickly position the cursor to any point on the screen.

If, while the terminal is in GIN state, the user strikes any key on the main keyboard (excluding the top row and CTRL, SHIFT, CAPS LOCK, SCROLL, and REPEAT) the terminal will automatically transmit a message to the host indicating the GIN cursor position. The format of this sequence is as follows:

## OPERATION

1. The character struck to initiate transmission of the message.
2. The most significant five bits of the X address, in the form P01XXXXX.
3. The least significant five bits of the X address, in the form P01XXXXX.
4. The most significant five bits of the Y address, in the form P01YYYYY.
5. The least significant five bits of the Y address, in the form P01YYYYY.
6. A CR character. Transmission of this character is optionally enabled by switch SW1 sections 1 and 2 (see Section 3.2).
7. A EOT character. Transmission of this character is optionally enabled by switch SW1 sections 1 and 2 (see Section 3.2).

P signifies the parity bit.

Once transmission of the message has started, the GIN cursor will disappear and the terminal will ignore all received characters until one of two events occur:

1. One of the following characters is received: BEL, BS, CR, HT, LF, US, GS, or VT.
2. One of the following character sequences is received: (ESC ETB) or (ESC FF) or (ESC X) or any valid sequence beginning (ESC :).

When one of the above is received the terminal will leave GIN state and enter Alpha state. The Alpha cursor will appear at the point last occupied by the GIN cursor. The practical effect of this is usually to ignore any characters echoed by the host computer until such time as the CR is echoed, which returns the terminal to normal Tektronix mode Alpha state operation. This means that the message can be received by a host computer operating in full duplex without any necessity for suspension of character echoing by the host.

The terminal also responds to the character sequence (ESC ENQ) while in GIN state. The effect is in all ways identical to that produced by striking a key except that the first character of the message (character struck) is not included. The message generated in response to (ESC ENQ) is thus one character shorter than that generated by striking a key.

## OPERATION

If, while in GIN state, the terminal receives a CR, or the sequence (ESC FF), or a global escape sequence changing the terminal display mode, or the local function key ERASE is struck, the terminal will exit GIN state without transmitting a cursor position report. While in GIN state the local function keys work normally; the programmable function keys, however, are disabled. The numeric keypad transmits no characters and can only be used for cursor position control.

**4.6.3.1 Pre-Positioning The GIN Cursor** - A Tektronix mode escape sequence is available for pre-positioning the GIN cursor. By default, when GIN state is entered the cursor appears at the same point on the screen that it occupied when the terminal was last in GIN state (or the center of the screen if GIN state has not been entered since power up or reset). The escape sequence (ESC X) changes the "remembered" GIN cursor position to that of the last vector end point. This sequence is only recognized while the terminal is in Tektronix mode. If the sequence is received while the terminal is in Tektronix mode Alpha state, then the "remembered" GIN cursor position is set to correspond to the current position of the lower left corner of the Alpha state cursor. If the sequence is received while the terminal is in Tektronix mode Graph state, the "remembered" GIN cursor position is set to correspond to the last specified vector end point.

This escape sequence can be used to pre-position the GIN cursor to the area of the screen that will be most convenient for the user of the program. The usual technique is to enter Tektronix mode Graph state, draw a dark vector to the desired point on the screen, issue the escape sequence (ESC X), and then issue the escape sequence (ESC SUB) to enter GIN state. The GIN cursor will then appear at the desired point.

### 4.6.4 Selective Erase Operations

Selective erase operations remove selected characters, dots, or lines from the screen without affecting the rest of the screen contents. There are three ways to perform selective erase operations:

1. **Black vectors:** As was discussed in paragraph 4.6.2.2, drawing a black vector erases all dots in its path. To draw a black vector it is only necessary to precede the vector end point specification with the control character NAK. In situations requiring the erasure of only a few vectors (rather than a long sequence of consecutive vectors) using black vectors is a very efficient technique.

## OPERATION

2. Area erase vectors: As was discussed in paragraph 4.6.2.3, area erase vectors erase all dots within a rectangular area of the screen. If the number of vectors and/or characters requiring erasure is not small, and they fall within a rectangular area, using an area erase vector is the most efficient technique.
3. Changing the data level to black: The escape sequence (ESC DEL) changes the data level to black. While the data level is black all alpha character and light vectors are drawn black; i.e., they remove dots from the screen. This applies to both Alpha state and Graph state. The usual technique for selectively erasing alpha characters is to set the data level to black and then "retype" the characters to be erased. It is also frequently convenient to erase a sequence of vectors by setting the data level to black and then "redrawing" the vectors. The data level does not affect the action of black vectors, area erase vectors, or area fill vectors.

The escape sequence (ESC a) resets the data level to white. The data level is also reset to white by the action of the ERASE key and by the erase screen escape sequence (ESC FF). The data level is always white when Tektronix mode is entered from a text mode.

### 4.6.5 Control Characters And Escape Sequences

The terminal responds to the following control characters when in Tektronix mode:

| <u>CHAR</u> | <u>CODE</u> (Hex) | <u>EFFECT</u>   |
|-------------|-------------------|---|
| BS          | 08                | Backspaces the cursor. Backspacing past the current margin causes wraparound.   |
| CR          | 0D                | Returns cursor to current margin. If the terminal is in Graph state, it is returned to Alpha state and the cursor appears at the current margin with the same Y position as the last vector end point. If no vectors were drawn in Graph state the margin will be the same as it last was in Alpha state, otherwise it will be cleared. If the terminal is in GIN state it is returned to Alpha state with the cursor at the left edge of the screen and at the same vertical position as the GIN state cursor. If Switch 402 section 3 on the Z19 logic board is set to "1" a LF will automatically be performed after the CR. |

## OPERATION

| <u>CHAR</u> | <u>CODE (Hex)</u> | <u>EFFECT</u>   |
|-------------|-------------------|---|
| LF          | 0B                | Line feed moves the cursor down 14 tekpoints. If the terminal is in Alpha state and the resulting position is below tekpoint 0Y, a margin switch (0 to 1 or vice versa) will occur and the Y position of the cursor will be set to 767. In Graph state the new Y position is computed modulo 1024, which can result in a Y position greater than 779 in some cases. If Switch 402 section 4 on the Z19 logic board is set to "1", a CR will automatically be performed after the LF. This CR will not return the terminal to Alpha state from Graph state or GIN state. |
| HT          | 09                | Tab spaces one character to the right (14 tekpoints). In Alpha state horizontal wraparound to the current margin will occur if the resulting position is greater than 1023X. This wraparound operation generates a LF. In Graph state the new X position resulting from a HT is computed modulo 1024.   |
| VT          | 0B                | Reverse line feed moves the cursor up 14 tekpoints. If the terminal is in Alpha state the cursor will not move up above tekpoint 767Y. In Graph state the new Y position resulting from a VT is computed modulo 1024, which can result in Y positions greater than 779 in some cases.   |
| GS          | 1D                | Leave Alpha state, enter Graph state. The current vector end point will be in the same position as the lower left corner of the Alpha cursor. GS is also recognized from the non-Tektronix modes of the terminal (except Transparent). In this case, the terminal enters Tektronix mode Graph state, the screen is cleared, and the current vector end point is set to 0X, 767Y. In all cases, the next vector drawn after a GS is dark. If GS is received while in Graph state the only effect is to make the next vector dark.  |
| US          | 1F                | Leave Graph state, enter Alpha state. The Alpha mode cursor will appear with its lower left corner at the last vector end point. If any vector drawing occurred in Graph state the margin will be cleared to margin 0. US has no effect if received in Alpha state or Gin state.  |
| NAK         | 15                | If received in Graph state, NAK causes the next vector to be drawn black. No effect in Alpha state.   |

## OPERATION

| <u>CHAR</u> | <u>CODE (Hex)</u> | <u>EFFECT</u>  |
|-------------|-------------------|--|
| FS          | 1C                | If received in Graph state, FS causes the next vector to be an area fill. No effect in Alpha state.  |
| RS          | 1E                | If received in Graph state, RS causes the next vector to be an area erase. No effect in Alpha state. |
| BEL         | 07                | 1000 HZ audible tone for 200 MS.   |

Note that cursor moving control characters (BS, CR, LF, VT, HT) are recognized in Graph state as well as Alpha state. Although there is no visible cursor to move, the current vector end point is moved just as the cursor would be. Since the current vector end point and the lower left corner of the cursor are always the same, there is no logical distinction between moving the cursor and moving the current vector end point.

The terminal also recognizes the following special escape sequences when in Tektronix mode. The first three sequences are recognized by all Tektronix mode states, while the remainder are recognized only in Alpha state or Graph state:

| <u>ESC SEQUENCE</u> | <u>EFFECT</u>   |
|---------------------|---|
| ESC FF              | Clears the screen, sets Alpha state, moves the cursor to home and sets the margin to MARGIN 0.  |
| ESC : Pn P          | This global escape sequence has the same affect as ESC FF when received in Tektronix mode. Parameter Pn is ignored.                                 |
| ESC ENQ             | Causes terminal/status and or current vector end point or crosshair cursor position to be transmitted (see paragraphs 4.6.1.3, 4.6.2.4, and 4.6.3). |
| ESC SUB             | Enables crosshair cursor (see paragraph 4.6.3).   |
| ESC DEL             | Causes all succeeding light vectors and alpha characters to be drawn black. The effect is cancelled by an (ESC a) or (ESC FF). See paragraph 4.6.4. |
| ESC a               | Cancels the effect of (ESC DEL).  |
| ESC ETB             | Activates the optional hardcopy device.   |
| ESC X               | Pre-positions the GIN cursor to the current vector end point (see paragraph 4.6.3.1).   |

## OPERATION

### 4.7 EASY-ENTRY GRAPHICS MODE

Easy-Entry Graphics mode can be entered by striking the local function key CTRL-SHIFT-WHITE or with the global escape sequence (ESC : 7 f). This mode allows vector drawing using a decimal notation suitable for direct keyboard entry. Although this notation is less efficient than that used by Tektronix mode Graph state operation (in terms of characters transmitted per vector drawn), it is sometimes easier to use. It is of greatest value when vectors are being entered directly from the keyboard or when the host computer does not have a library of Tektronix compatible plotting subroutines available.

Easy-Entry Graphics mode uses single letter commands which are preceded, when necessary, by vector end points specified as decimal numbers. Vector end points are specified in terms of dots on the screen rather than "tekpoints"; thus valid X addresses range from 0-511 and valid Y addresses range from 0-249. The origin of this coordinate system is at the lower left hand corner of the screen. The address of a new vector end point may be specified either as an absolute location in the coordinate system or relative to the current vector end point. Commands have the following syntax:

Ps c

Ps is a parameter string of the form:

[+-]X [+-]Y

where X and Y are integers up to three digits in length specifying the desired X and Y screen coordinates. If an X or Y coordinate is preceded by a plus or minus sign it indicates that the number is to be regarded as an offset relative to the current vector end point. The integers specifying the X and Y coordinates must be separated by a space or a sign character (+-). The brackets are not part of the syntax and only indicate that the enclosed characters are optional. The letter c signifies one of the following lower case command characters:

#### LETTER

#### MEANING

- |   |   |
|---|---|
| o | Reset the current vector end point to the coordinate system origin, 0X 0Y (lower left screen corner). No parameter string is accepted or required by this command. Equivalent to a Tektronix mode dark vector drawn to point 0X 0Y. |
| p | Position current vector end point. The current vector end point is moved to the location specified by Ps. Equivalent to a Tektronix mode dark vector drawn to the point specified by Ps.  |



## OPERATION

### LETTER

### MEANING

- v Draw a vector from the current vector end point to that point specified by Ps. Equivalent to a light vector.
- u Undraw vector. Removes all dots in the path of a vector from the current vector end point to the location specified by Ps. Equivalent to a black vector.
- d Draw a single dot at the location specified by Ps.
- w Set data level to white. All subsequent vectors and characters will be formed by turning on dots on the screen. This effects only 'v' and 'd' commands and characters drawn after an 'a' command. The data level is set to white when Easy-Entry mode is first entered from a text mode, by the action of the ERASE key, and by the action of the escape sequence (ESC FF). No parameter string is accepted or required by this command.
- b Set data level to black. All subsequent vectors and characters will be drawn by turning off dots on the screen. This effects only 'v' and 'd' commands and characters drawn after an 'a' command. This allows these commands to be used as selective erase operations. No parameter string is accepted or required by this command.
- f Fill the rectangular area whose diagonal is specified by the current vector end point and Ps.
- e Erase the rectangular area whose diagonal is specified by the current vector end point and Ps.
- r Report current vector end point. The terminal will transmit the current end point address in the form:  
XXX YYY CR  
Where XXX and YYY are three digit integers, with leading zeros if needed, separated by a single space. No parameter string is accepted or required by this command.
- a Enter Tektronix mode Alpha state. If Ps is specified the alpha cursor will be placed at that point, otherwise it will be placed at the current vector end point.

## OPERATION

Commands are executed as soon as the command character is recognized. The CR character may be entered at any point to terminate a sequence. This will cause any existing partially complete sequence to be discarded. Spaces are legal at any point in the sequence except between digits of an integer. Any commands containing illegal coordinates are ignored.

As an example, the following command sequence draws a box around the edge of the display area:

```
0 0 p 0 249 v 511 249 v 511 0 v 0 0 v
```

The equivalent sequence, using relative addressing, is as follows:

```
0 0 p +0 +249 v +511 +0 v +0 -249 v -511 +0 v
```

The differences between the two addressing methods can be seen by comparing the two examples. The first line drawn, from point (0, 0) to point (0, 249), requires the same parameters in both examples. In this case the relative displacement (from point (0, 0) to point (0, 249)) is the the same as the address of the new vector end point. In the case of the second line drawn, from point (0, 249) to point (511, 249), the parameters are different. In this case the relative displacement along the X axis is 511 points (511-0) while the relative displacement along the Y axis is 0 (249-249). Thus with relative addressing the parameter string for this vector end point is +511, +0 and with absolute addressing it is 511, 249.

It is also possible to mix relative and absolute address specifications. Each coordinate within each parameter string can be either absolute or relative.

Although Easy-Entry mode does not have a unique Alpha state for drawing characters, Tektronix mode Alpha state can be entered by using the 'a' command. To return to Easy-Entry mode from Tektronix mode Alpha state use the appropriate global escape sequence (ESC : 7 f), or, if commands are being entered directly from the keyboard, use the local function key CTRL-SHIFT-WHITE. When returning to Easy-Entry mode from Tektronix mode Alpha state the current vector end point will be positioned at the point last occupied by the lower left corner of the alpha cursor.

The ERASE key, or the escape sequence (ESC FF), will erase the screen and then cause the terminal to enter Tektronix mode Alpha state (with the data level set to white). The local function key CTRL-SHIFT-F5, or the global escape sequence (ESC : E), will erase the screen and leave the terminal in Easy-Entry mode without affecting the data level.

## OPERATION

### 4.8 GRAPHICS SCROLLING

When the terminal is in any of the graphics modes the display can be scrolled horizontally either left or right by a global escape sequence. The format of the escape sequence is:

ESC : Pb;Pt;Pd S

Parameters Pb and Pt specify the bottom and top of the horizontal region that is to be scrolled. These parameters are specified in screen dot positions, thus the bottom of the screen corresponds to a parameter value of 0 and the top of the screen corresponds to a value of 249 (see Section 4.6 for information on converting tekpoint coordinates to screen dot positions). If the parameters are not in this range, or if Pb is greater than Pt, the sequence will be ignored. Parameter Pd specifies direction; Pd=0 requests scrolling to the left and Pd=1 requests scrolling to the right.

Each time the escape sequence is received the selected region is scrolled eight dots (16 tekpoints) in the specified direction. The portion of the image that scrolls off the edge is lost and the portion that scrolls onto the opposite side is blank. This feature can be employed to simulate a chart recorder or moving histogram by first scrolling the screen and then displaying the next eight new data points on the portion of the screen that was blanked by the scrolling operation. After eight new data points have been displayed the screen is then scrolled again, etc.

### 4.9 TRANSPARENT MODE

In Transparent mode the terminal will display all received characters including control characters. Transparent mode can be entered by striking the local function key CTRL-SHIFT-ESC and exited by striking CTRL-SHIFT-ESC again or any other mode changing local function key. Note that the terminal cannot exit Transparent mode in response to an escape sequence from the host computer because the terminal does not recognize escape sequences or control characters (except CR) while in this mode.

Control characters are displayed as the corresponding printing character in reverse video; control characters thus appear as black characters in white character cells. The only exception to this rule is the control character DEL, which is displayed as a solid white character cell.

The only control character to which the terminal responds while in this mode, other than by simply displaying it, is CR. The received CR is first displayed (as a control-M) and the cursor is then returned to the left margin and moved down one line. While in transparent mode the

## OPERATION

terminal always performs automatic line wraps; that is, when the cursor reaches the 80th (or 132nd) column it automatically returns and displays the next character at the beginning of the next line.

### 4.10 PROGRAMMABLE KEYS

The keys F1, F2, F3, F4, F5, BLUE, RED, and WHITE can be programmed to transmit strings of characters when struck. In fact, each of these keys can actually be programmed to transmit two different strings: one when the key is struck while holding down SHIFT and another when the key is struck alone.

The keys are programmed either from Set-up mode (see Section 4.3) or by the use of escape sequences. The global escape sequence for programming keys has the general form:

ESC : Pn p DELIMITER text DELIMITER

The parameter Pn specifies which key is to be programmed (see Table 4.2). The word DELIMITER refers to a character used to delimit the text string. This can be any character and will function like a parenthesis: the first occurrence of the character marks the beginning of the text string and the second occurrence marks the end. Obviously the DELIMITER character must be chosen so as not to be the same as any character that appears in the text string. The text string can be up to 128 characters in length and may contain any and all ASCII characters except NUL. Any characters in excess of 128 will be discarded. When the programmed key is struck, it will transmit the exact same string of characters as was between the DELIMITER characters. Programmed keys may be erased by use of the global escape sequence:

ESC : Ps e

The parameter Ps specifies the key, or keys, to be affected (see Table 4.2). If more than one key is being specified the key numbers should be separated by semicolons. All programmable keys may be erased simultaneously by using the escape sequence:

ESC : e

Erasing a programmable key resets it to the default Z19 escape sequence for the key (depending on the current escape mode of the terminal, either an ANSI or a ZENITH sequence will be used).

## OPERATION

TABLE 4.2

Identifying Parameter for each Programmable Key

| Key   | Unshifted | Shifted |
|-------|-----------|---------|
| F1    | 1         | 9       |
| F2    | 2         | 10      |
| F3    | 3         | 11      |
| F4    | 4         | 12      |
| F5    | 5         | 13      |
| BLUE  | 6         | 14      |
| RED   | 7         | 15      |
| WHITE | 8         | 16      |

As Table 4.2 shows, the shifted and unshifted behavior of each of the function keys can be programmed separately, giving the user 16 possible programmable functions. Note, however, that the total of all the text strings assigned to programmable function keys cannot exceed 512 characters. If an attempt to program a key causes the 512 character limit to be exceeded, then the key will be programmed with the portion of the character sequence for which memory was available. The portion of the sequence that arrives after the 512 character limit is reached will be discarded.

In order for programmable keys to transmit their programmed character sequences rather than their native default character sequences, the Programmable Keys Active state of the terminal must be set, either from Set-up mode (see Section 4.3) or with the proper global escape sequence (ESC : 5 h). There is a corresponding global escape sequence (ESC : 5 l) to clear this state; the state is also cleared by a terminal reset operation. Note that clearing this state does not erase programmed character sequences stored in the terminal's memory nor does it prevent the storage of new character sequences; it only affects the action that occurs when programmed function keys are struck. Also note that if the Programmable Keys Active state is set, but a particular function key has not been programmed, then that function key will still transmit its native default character sequence when struck.

The native Z19 sends the same escape sequence for a given function key without regard to whether or not SHIFT was depressed. It is possible for the user to preserve the native function of a key by programming only its shifted counterpart. For example, the user could leave the normal sequence associated with the F1 key intact while assigning the string "directory/date" to SHIFT-F1 by using the following global escape sequence:

```
ESC : 9 p *directory/date CR*
```

## OPERATION

Note that \* was used as the DELIMITER character and that a CR was included to terminate the command for the host computer. Since the character sequence for a programmable key can be up to 128 characters long, and since CR is a legal character in the sequence, it is also possible to imbed two or more host command lines in a single character sequence. For example, the following global escape sequence would program SHIFT-F1 to transmit the command "show system" followed by the command "show devices":

```
ESC : 9 p *show system CR show devices CR*
```

Such a technique will only work with host computers that allow typeahead.

### 4.11 REVERSE SCREEN STATE

The terminal has a Reverse Screen state, in which the video level of all screen displays is reversed; that is, characters and graphics will appear as black on a white background. This state affects all information displayed in all modes of operation. Reverse Screen state can be entered by striking the local function key CTRL-F4. This key combination actually toggles the state; striking it again will return the terminal to Normal Screen state. Reverse Screen state can also be set with the global escape sequence (ESC : 7 h) and cleared with the sequence (ESC : 7 l).

### 4.12 ADDITIONAL NEW FEATURES

#### 4.12.1 Scroll Key

The scroll key now has two additional functions. When Holdscreen operation is not enabled, the key operates in a manner similar to that found on VT100 compatible terminals. The key toggles, alternately stopping or starting the scrolling of lines onto the screen each time it is struck. Scrolling is stopped by sending a DC3 (CTRL-S, "XOFF") to the host computer and stopping the removal of characters from the terminal input buffer. When scrolling is started by striking the key again, a DC1 (CTRL-Q, "XON") is sent to the host computer and simultaneously the terminal resumes removing characters from the input buffer for display. This feature is disabled if Holdscreen operation is enabled.

If SHIFT-SCROLL is struck, Holdscreen operation is automatically enabled (for further information on Holdscreen operation see the VIDEO TERMINAL OPERATION MANUAL supplied by Zenith. There is a switch option to enable/disable this feature, see Section 3.2). This provides a

## OPERATION

convenient way to enable Holdscreen operation with a single key stroke, rather than using an escape sequence or first entering Set-up mode. Holdscreen operation can likewise be conveniently disabled by striking CTRL-SCROLL. When the Holdscreen operation is enabled the scroll key functions exactly as it does on a Z19 without GRAPHICS-PLUS installed.

One peculiarity of Holdscreen and scroll key operation should be noted. Some "local" keys on the keyboard, such as CTRL-ERASE, CTRL-SHIFT-ERASE, and the keys of the numeric keypad used in conjunction with CTRL, function by placing the resulting characters in the terminal input buffer just as though they had arrived from the host computer. The same thing is true for all characters typed when the terminal is off line. When scrolling has been stopped by the scroll key or a Holdscreen operation the terminal stops removing characters from the input buffer for processing. Thus if one of these local keys is struck while scrolling is suspended, or the terminal is taken off-line and characters are struck on the keyboard while scrolling is suspended, the resulting terminal action will not occur until after scrolling is resumed. This does not apply to the local functions of the F1-F5 and BLUE, RED, WHITE, and ESC keys. Note that the scroll key stops/starts the removal of characters from the input buffer even when the terminal is off-line.

### 4.12.2 Programmable Tab Stops

In addition to the normal fixed tab stops at eight column intervals, the terminal allows the user to set tab stops at any and all column positions. In order for user-defined tab stops to be in effect, the user must set the User Tab Stops Active state, either from Set-up mode (see Section 4.3) or with the proper global escape sequence (ESC : 2 h). When the User Tab Stops Active state is set, the native fixed position tab stops are disabled and the tab stops programmed by the user become active. There is a corresponding escape sequence (ESC : 2 l) that clears this state and thus restores the native fixed position tab stops. Note that clearing the User Tab Stops Active state does not actually delete user programmed tab stop positions from the terminal memory; if the state is re-enabled any previously programmed tab stops will again become active. Also note that programmable tab stops can be set and cleared in terminal memory whether or not the state is active.

Programmable tab stops can be set from Set-up mode, or by using a global escape sequence of the form:

ESC : Ps s

The parameter Ps indicates the column, or columns, at which the stops are to be set. Columns are numbered 1 thru 132; if Ps=0 (or Ps is omitted) then a stop is set at the current cursor position. If multiple stops are being specified, the column numbers should be separated by

## OPERATION

semicolons. For example, the following escape sequence sets tab stops at columns 5, 15, 25, 50, 60, and 70:

```
ESC : 5;15;25;50;60;70 s
```

Programmed tab stops can be cleared by using a global escape sequence of the form:

```
ESC : Ps c
```

The meaning of the parameter Ps is the same as explained above with one exception: if the parameter Ps is omitted, all programmed tab stops will be cleared.

### 4.12.3 Back Tab Sequence

The terminal responds to a back tab sequence (ESC TAB) by moving the cursor left until a tab stop is encountered.

### 4.12.4 Smart Shift Key

The SHIFT keys now have an additional function not native to the Z19. If the SHIFT key is held down while the CAPS LOCK key is already in the depressed position, then any alphabetic key struck will produce a lower case rather than an upper case character. There is no effect on non-alphabetic keys. This feature can be disabled from Set-up mode (see Section 4.3) or by setting switch SW1 section 3 to "0" (see Section 3.2). There is also a global escape sequence to enable this feature (ESC : 9 h) and a corresponding global escape sequence to disable it (ESC : 9 l).

### 4.12.5 Smart Auto Wrap

If Smart Auto Wrap is enabled, the behavior of Auto Wrap (a standard Z19 feature) is modified. If Auto Wrap is not also enabled then enabling Smart Auto Wrap has no effect. The effect of Smart Auto Wrap is twofold:

1. Smart Auto Wrap prevents double spacing following lines that are exactly one screen width (80 or 132 characters) in length. Automatic line wrap (the terminal inserts a CR LF) still occurs immediately after a character is placed in the last column on the screen. However, if this character is followed directly by a CR, LF or CR LF combination, the CR and/or LF are ignored.



## OPERATION

2. Smart Auto Wrap allows "backwards" automatic wraps. If the cursor is backspaced past the left edge of the screen it will wrap to the end of the preceding line. This allows the cursor to backspace thru multiple lines of text. Note that the cursor will not wrap "backwards" past the top line of the screen.

Smart Auto Wrap is enabled by default. It can be disabled or enabled from Set-up mode. It can also be disabled with the global escape sequence (ESC : 12 l) and enabled with the global escape sequence (ESC : 12 h).

### 4.12.6 Screen Timeout

The screen timeout feature blanks the screen of the terminal if no characters are received, or keys struck, for a period of ten minutes. This extends the life of the CRT phosphor. The screen is automatically unblanked if a character is received or a key struck (except for the right SHIFT key and the RESET key). The left SHIFT key is a good choice since it will be recognized without actually producing a character. If this feature is not desired it can be disabled by a power-up switch, from Set-up mode, or by using a global escape sequence. Setting switch SW1 section 8 to the "1" position disables screen timeout. The global escape sequence (ESC : 4 h) also disables screen timeout. The global escape sequence (ESC :4 l) enables screen timeout.

### 4.12.7 Transmit Cursor Line

If RETURN is struck while CTRL is depressed, the terminal will transmit the entire line on which the cursor is located. The line is transmitted without trailing blanks and with a terminating CR character. Transmission will occur even if the terminal is in the Off-line state. The global escape sequence (ESC CR) has the same effect. CTRL-RETURN is useful for re-transmitting a complicated command line to the host. A complex command containing a simple error can be edited before re-transmission by taking the terminal off-line and using the keypad insert and delete character keys. This same technique can be employed for editing program lines when the host computer is running a BASIC language interpreter.

If graphics characters or reverse video characters are encountered in the line then appropriate escape sequences are also transmitted, just as in the case of the Transmit Status Line and Transmit Page functions. For more on this subject see the explanation of the Transmit Status Line and Transmit Page functions found in the VIDEO TERMINAL OPERATION MANUAL supplied by Zenith.

## OPERATION

### 4.12.8 Transmission Speed Limiting

If transmission speed limiting is enabled, the terminal's transmission rate will be limited to 60 characters per second. Note that this does not affect the baud rate, only the rate at which characters are removed from the output buffer and transmitted. This feature is useful with host computers that are not capable of handling high speed bursts of characters from terminals. This feature can be enabled from Set-up mode (see Section 4.3), by setting switch SW1 section 4 to the "1" position (see Section 3.2), or with the global escape sequence (ESC : 1 h). The sequence (ESC : 1 l) disables this feature.

### 4.12.9 Definable Scrolling Region

The terminal now has a definable scrolling region. Both the top and bottom lines of the screen region in which text scrolls can be defined. By default, the scrolling region is set to include the entire screen (lines 1:24 in 24 line mode, lines 1:49 in 49 line mode). When the scrolling region is set to include less than the entire screen, text that scrolls above the top line of the defined region is discarded; the contents of the screen not included in the scrolling region, and the contents of off-screen memory, are not affected.

The scrolling region can be defined by using Set-up mode (see Section 4.3) or by the use of escape sequences. Both a global escape sequence and an ANSI escape sequence are available. The global escape sequence has the following form:

```
ESC : Pt;Pb r
```

The ANSI escape sequence has the form:

```
ESC [ Pt;Pb r
```

In both cases the parameter Pt specifies the top line of the scrolling region and Pb specifies the bottom line. Whenever the scrolling region is re-defined by an escape sequence the cursor is automatically moved to the home position.

Whenever a new text mode format is specified, either by a local function key or an escape sequence, the scrolling region is automatically redefined to include the entire screen.

## OPERATION

4.12.9.1 Relative Vs. Absolute Origin Mode - Escape sequences that include a line number as a parameter may be affected by the scrolling region definition, depending on whether the terminal is in Relative or Absolute origin mode. When the terminal is in Absolute Origin mode (the default) line 1 is considered to be the top of the screen. When the terminal is in Relative Origin mode line 1 is the top line of the currently defined scrolling region; i.e., the line address is relative to the top of the scrolling region rather than the top of the screen.

There are two escape sequences that set Relative Origin mode, an ANSI sequence (ESC [ ? 6 h) and a global sequence (ESC : 16 h). Absolute Origin mode, the default, can also be set by an ANSI sequence (ESC [ ? 6 l) and a global sequence (ESC : 16 l).

Relative Origin mode affects both direct cursor addressing escape sequences and cursor position reporting escape sequences. This includes ZENITH, ANSI, and global escape sequences. The affected ZENITH escape sequences are:

ESC Y Pr Pc (direct cursor addressing)  
ESC n (cursor position report)

The affected ANSI sequences are:

ESC [ Pr;Pc H (direct cursor addressing)  
ESC [ Pr;Pc f (direct cursor addressing)  
ESC [ 6 n (cursor position report)

The affected global sequences are:

ESC : Pr;Pc H (direct cursor addressing)  
ESC : 6 n (cursor position reporting)

The ZENITH, ANSI and Global escape sequences that erase the entire screen are also affected by the defined scrolling region when Relative Origin mode is set. In this case the cursor is returned to the beginning of the scrolling region when the screen is cleared rather than to the upper left corner of the screen. There is one exception to this rule: if the terminal is in ANSI escape mode, and EDT compatibility is set (see paragraph 4.12.10), then clearing the screen does not affect the position of the cursor. This allows compatibility with VT100 type terminals. The affected escape sequences are:

ESC : 1 P (Global clear screen)  
ESC E (ZENITH clear screen)  
ESC [ 2 J (ANSI clear screen)

## OPERATION

### 4.12.10 EDT Compatibility

The terminal will operate with the DEC EDT screen editor as either a VT52 or a VT100 compatible terminal. The advantage of operating with EDT as a VT100 is that the 132 column and reverse video capabilities of the terminal can be utilized by the editor in this mode. In order for the terminal to appear as a VT100, the EDT Compatibility state of the terminal must be set. This state only affects the terminal if it is also in ANSI escape mode; it does not affect the terminal's behavior in ZENITH escape mode. The effect on the terminal in ANSI escape mode is twofold: It changes the way the terminal responds to receiving some escape sequences (see paragraph 4.13.1.1), and it changes the alternate escape sequences generated by the numeric keypad.

The alternate escape sequences generated by the numeric keypad when EDT Compatibility and ANSI escape mode are in effect are shown in Table 4.3. The correspondence between keys and escape sequences is intended to provide a reasonably convenient keypad layout for using EDT which is largely consistent with the VT52 mode of operation. The EDT function of each key is shown in Figure 4.2.

OPERATION

FIGURE 4.2

"EDT Compatible" Keypad Functions

|                |             |                           |
|----------------|-------------|---------------------------|
| "Gold"<br>BLUE | Help<br>RED | Fndnxt<br>WHITE<br><Find> |
|----------------|-------------|---------------------------|

Unshifted

|                          |                        |                          |
|--------------------------|------------------------|--------------------------|
| Page<br>7<br><Command>   | Fndnxt<br>8<br><Find>  | Del W<br>9<br><Und W>    |
| Advance<br>4<br><Bottom> | Backup<br>5<br><Top>   | Del C<br>6<br><Und C>    |
| Word<br>1<br><Chgcase>   | Eoln<br>2<br><Del Eol> | Cut<br>3<br><Paste>      |
| Line<br>0<br><Open Ln>   | Select<br>.<br><Reset> | Enter<br>enter<br><Subs> |

Shifted

|                        |                        |                          |
|------------------------|------------------------|--------------------------|
| Sect<br>7<br><Fill>    | Up<br>8                | Append<br>9<br><Replace> |
| <---<br>4              | Backup<br>5<br><Top>   | --><br>6                 |
| Char<br>1<br><Specins> | Down<br>2              | Del Ln<br>3<br><Und Ln>  |
| Line<br>0<br><Open Ln> | Select<br>.<br><Reset> | Enter<br>enter<br><Subs> |

White is the same as the Unshifted 8 key (Fndnxt / Find).  
F1 is the same as the Shifted 3 key (Del Ln / Und Ln).

4.12.11 VT100 Character Set

The terminal now has two character sets, the standard Zenith character set and a replica of the VT100 character set. These two character sets differ greatly in their special graphics symbols. The text character fonts in the two set are quite similar, but not exactly the same. The character set can be selected by a power up switch, from Set-up mode, or by an escape sequence. If switch SW1 section 5 is set to "0" the VT100 character set will be selected by default. If it is set to a "1" the standard Zenith character set will be selected by default. The global

## OPERATION

escape sequence (ESC : 10 h) selects the VT100 character set and the sequence (ESC : 10 l) selects the standard Zenith character set.

Appendix E shows the graphics characters available in the VT100 character set.

### 4.12.12 Status Line Addressing

There is a global escape sequence for moving the cursor directly to the status line. By using this escape sequence programs can address the status line located at the bottom of the screen without having to consider whether the terminal is currently in a 24 line or a 49 line text display mode. This sequence also automatically enables the status line. The format of this sequence is:

ESC : Pn L

The parameter Pn specifies the column number to which the cursor will be positioned on the status line. If Pn is omitted the cursor will be positioned to the left hand end of the status line.

Before the cursor is positioned a "save cursor position" operation (equivalent to ESC 7, see Section A.7) is automatically performed. Thus, when operations on the status line are complete, the cursor can be returned to its former position on the screen by use of the "restore cursor position" escape sequence. (ESC 7 - "save cursor position" and ESC 8 - "restore cursor position" work in both the Zenith and ANSI escape modes. See Section A.7.)

### 4.12.13 Output Synchronization

If output synchronization is enabled the terminal will recognize DC1 (CTRL-Q, "XON") and DC3 (CTRL-S, "XOFF") control characters when they are received from the host. Receipt of a DC3 character will cause the terminal to cease transmitting characters from its output buffer to the host. Receipt of a DC1 character re-enables transmission of characters by the terminal. When the terminal is connected to a host computer that supports this protocol, output synchronization can be used to prevent host computer input buffer overflows. If output synchronization is not enabled the terminal ignores received DC1 and DC3 characters unless it is in Transparent mode, in which case they are displayed.

Output synchronization is disabled by default. Output synchronization can be enabled from Set-up mode or by using the global escape sequence (ESC : 17 h). It can be disabled from Set-up mode or by using the global escape sequence (ESC : 17 l).

## OPERATION

If output synchronization is enabled, and transmission has been stopped by receipt of a DC3 character, entering and exiting Set-up mode will re-enable transmission of characters by the terminal. This serves as a method for "unlocking" the terminal if the host computer fails to send the needed DC1 character.

### 4.12.14 Variable Length Key Click

There is a global escape sequence for generating "key click" tones of variable length. The form of this escape sequence is:

ESC : Pn T

The value of parameter Pn controls the duration of the sound generated when the escape sequence is received. The value of Pn must be in the range 0-255. Although the relationship between Pn and the length of the sound generated is not linear, it is generally true that larger values of Pn generate longer sounds.

## 4.13 ESCAPE SEQUENCES

Escape sequences are special character sequences that begin with the control character ESC. Many features of the Z19 can be controlled by escape sequences and the Z19 transmits escape sequences when certain keys are struck. The Z19 recognizes and transmits two rather different sets of escape sequences, one when the terminal is in ANSI escape mode and another when the terminal is in ZENITH (or Heath) escape mode. GRAPHICS-PLUS considerably expands the terminal's repertoire of escape sequences. The new escape sequences mostly fall into two categories: ANSI compatible escape sequences and global escape sequences. The following paragraphs are intended to briefly summarize these additions; a complete list of escape sequences can be found in Appendix A.

### 4.13.1 ANSI Compatible Escape Sequences

These are new escape sequences that are recognized when the terminal is in ANSI escape mode. All of the new sequences in this category are the same as those recognized by VT100 compatible terminals. Included in this category are sequences for setting 80 and 132 column screen formats, entering and exiting reverse screen state, and setting tab stops. Also included in this category are some "ignored" VT100 escape sequences; the terminal recognizes these sequences as valid but takes no action upon receiving them. These are included so that the terminal can be operated in ANSI escape mode with a host computer configured for

## OPERATION

VT100 terminals without unrecognized VT100 escape sequences showing up as garbage characters on the screen.

4.13.1.1 EDT Compatibility - In the case of the escape sequence for defining the scrolling region, a conflict exists between the sequence used by the DEC VT100 and the ANSI sequence used by Zenith for setting terminal baud rate. This conflict is resolved by setting or clearing EDT compatibility state. When EDT compatibility state is set the VT100 escape sequence for defining the scrolling region is properly recognized and no ANSI sequence for setting baud rate exists (there is a global escape sequence available to accomplish the same task). All other ANSI escape sequences are recognized just as they are in ANSI escape mode with the EDT compatibility state cleared. When the state is cleared the terminal correctly recognizes the Zenith ANSI sequence for setting baud rate so long as the sequence contains only one parameter (the normal case). If the sequence contains no parameters or two parameters it is assumed to be a VT100 sequence for defining the scrolling region. The recognition of sequences in ZENITH escape mode is not affected by this state.

EDT compatibility also affects the way the cursor is handled during clear screen operations. Unlike the Z19, the DEC VT100 terminal does not return the cursor to the "home" position when the screen is cleared. Thus when EDT compatibility is set the terminal does not move the cursor when a clear screen operation is performed. See paragraph 4.12.9.1 for a list of the clear screen escape sequences that are affected.

Setting EDT compatibility state affects ANSI escape mode in one other respect. In order to facilitate use of the DEC EDT screen editor in VT100 mode, the alternate escape sequences transmitted by the numeric keypad have been redefined so that the functional layout of the keypad is convenient and largely consistent with the VT52 mode of operation. This re-definition occurs only if the terminal is in ANSI escape mode, EDT compatibility state is set, and the keypad is set to alternate mode. The escape sequences generated by the keypad under these circumstances are shown in Table 4.3.



## OPERATION

TABLE 4.3

### EDT Compatible Keypad Escape Sequences

| Key   | Unshifted | Shifted             |
|-------|-----------|---------------------|
| 9     | Esc O m   | Esc O y             |
| 8     | Esc O R   | Esc [ A (Esc O A) * |
| 7     | Esc O w   | Esc O x             |
| 6     | Esc O l   | Esc [ C (Esc O C) * |
| 5     | Esc O u   | Esc O u             |
| 4     | Esc O t   | Esc [ D (Esc O D) * |
| 3     | Esc O v   | Esc O S             |
| 2     | Esc O r   | Esc [ B (Esc O B) * |
| 1     | Esc O q   | Esc O s             |
| 0     | Esc O p   | Esc O p             |
| .     | Esc O n   | Esc O n             |
| Enter | Esc O M   | Esc O M             |

\* Sequences in parenthesis will be sent if ANSI cursor keys are set to 'alternate' mode.

ANSI escape sequences, EDT compatibility, and Alternate Keypad mode must be set for these sequences to be generated.

#### 4.13.2 Global Escape Sequences

Most of the new escape sequences are in this category. A global escape sequence is one that is recognized in both ZENITH and ANSI escape modes. Included in this category are sequences to change to any terminal mode, alter most controllable attributes, and program the programmable function keys and tab stops.

#### 4.13.3 Other New Escape Sequences

A few new escape sequences do not fit into the above categories. These include the escape sequences that are part of the Tektronix protocol (see Section 4.6.5), the back tab escape sequence (see paragraph 4.12.3), and the transmit cursor line escape sequence (see paragraph 4.12.7). These latter two sequences are "global" in the sense that they are recognized in both terminal escape modes, but do not have the same form as other global escape sequences.

## OPERATION

### 4.14 INPUT AND OUTPUT BUFFERS

The terminal's internal input and output buffers are now larger and functionally more complicated than in the native Z19. The input buffer (the buffer used to hold characters arriving from the host computer) is now 1024 characters, rather than 128. If the input buffer becomes nearly full (more than 960 characters), a DC3 (CTRL-S, "XOFF") is automatically transmitted to request that the host stop sending characters. An additional DC3 character is automatically transmitted each time a character is received while the input buffer contains more than 960 characters. When the buffer becomes less full (less than 896 characters), a DC1 (CTRL-Q, "XON") is transmitted to request that the host resume transmission. If the input buffer becomes completely full, arriving characters are discarded and the terminal bell is sounded.

The output buffer (used to hold characters awaiting transmission) is now 128 characters long rather than 32. If Output Synchronization is enabled and the terminal receives a DC3 character from the host then transmission of characters by the terminal will stop, and remain stopped, until such time as time a DC1 is received. Any characters struck on the keyboard will be stored in the output buffer until such time as transmission resumes. If a programmed key is struck then the entire programmed character sequence is stored in the output buffer. If the output buffer becomes full then the terminal bell will be sounded and all characters produced by the struck key will be discarded.

If the terminal is operating in half-duplex then characters from the keyboard must be stored in both the input buffer and the output buffer. It is thus possible, although unlikely, to get an overfull condition in either buffer as a result of a key, or programmed key, being struck while in half-duplex.

## CHAPTER 5

### TROUBLESHOOTING

#### 5.1 TROUBLESHOOTING PROCEDURES

If the terminal fails to operate properly, the following procedure should be followed to isolate the cause of the failure:

1. Run the internal diagnostic programs to check for memory failures as described in Section 5.2. If the diagnostics indicate a failing memory device it is probable that this is the cause of the problem. Since all memory devices are installed in sockets it is possible for users with available spares to replace failing memory devices in the field. If the failing memory device is a program ROM, a replacement ROM may be obtained from Northwest Digital Systems. If the internal diagnostics execute without detecting a memory failure, or fail to execute as described in Section 5.2, proceed to step 2.
2. Determine whether the fault is in the GP-19 hardware or in the Z19 terminal. To do this, first unplug the terminal power cord and then remove the GP-19 hardware by reversing the installation procedure described in Chapter 3. Be sure to remember to return the Z80 microprocessor IC and the 6845 CRT controller IC to the correct positions on the Z19 logic board. Take extreme care not to damage the pins on these ICs while moving them. If the terminal is a Z19-CN, apply insulating tape to the connector on the end of the cable that supplied line voltage to P1 on the auxiliary power supply. Take care to make certain that the connector pins are covered so they can not come in contact with other parts of the terminal. Turn on the terminal and test it for normal functioning. If the Z19 terminal fails to work correctly without the GP-19 hardware installed then the Z19 terminal is at fault and should be repaired. If the Z19 terminal works correctly without the GP-19 hardware installed, then the GP-19 hardware should be returned to Northwest Digital Systems for repair.

## TROUBLESHOOTING

### 5.2 USING THE SELF-DIAGNOSTICS

GRAPHICS-PLUS has internal diagnostic programs for checking the integrity of all program RAMs and ROMs in the terminal. If a memory failure occurs, these programs will identify the integrated circuit at fault by displaying a code in the LEDs on the upper left corner of the board.

To run the memory diagnostics, turn on the terminal while holding down the left SHIFT key. The terminal will beep once (as it normally does) and then make a "tick" sound (a short beep) three seconds later; do not release the SHIFT key until the "tick" occurs. The diagnostics will then run for about 17 seconds; during this time all the LEDs will be on. If no failures are found, the terminal will proceed with its normal power up sequence at the end of the 17 second period. If a failure was found, the terminal will beep and display the code for the failing device in the LEDs. If more than one failing device was found, the terminal will cycle thru the failures, beeping each time it displays a new error code, at a rate of about one every three seconds. The user can force the terminal to exit the diagnostic routine, and proceed with the power up sequence, by striking SHIFT-RESET (right shift key). Table 5.1 shows the correspondence between the LED error codes and device part numbers.

## TROUBLESHOOTING

### TABLE 5.1

#### Power-Up Diagnostic LED Error Codes

LED Error Code  
(1 = LED ON)

D4 D3 D2 D1

|   |   |   |   |                    |
|---|---|---|---|--------------------|
| 0 | 0 | 0 | 0 | Video Ram D0 - U35 |
| 0 | 0 | 0 | 1 | Video Ram D1 - U34 |
| 0 | 0 | 1 | 0 | Video Ram D2 - U33 |
| 0 | 0 | 1 | 1 | Video Ram D3 - U32 |
| 0 | 1 | 0 | 0 | Video Ram D4 - U31 |
| 0 | 1 | 0 | 1 | Video Ram D5 - U30 |
| 0 | 1 | 1 | 0 | Video Ram D6 - U29 |
| 0 | 1 | 1 | 1 | Video Ram D7 - U28 |

|   |   |   |   | Z19 H19 WH19 | Z19-CN Z19-HN H19-A          |                              |
|---|---|---|---|--------------|------------------------------|------------------------------|
|   |   |   |   | -----        | -----                        |                              |
| 1 | 0 | 1 | 0 | 2114s        | U467 or U469<br>U468 or U470 | U409 or U411<br>U408 or U410 |
| 1 | 0 | 1 | 1 | 2112s        | U424 or U425                 | U438 or U439                 |
| 1 | 1 | 0 | 0 | ROM #4 - Ext |                              |                              |
| 1 | 1 | 0 | 1 | ROM #3 - U24 |                              |                              |
| 1 | 1 | 1 | 0 | ROM #2 - U25 |                              |                              |
| 1 | 1 | 1 | 1 | ROM #1 - U26 |                              |                              |

## CHAPTER 6

### CIRCUIT DESCRIPTION

#### 6.1 INTERCONNECTION WITH THE Z19

The GP-19 logic board connects to the existing terminal via either three or four cables (four in the case of a Z19-CN). Logic signals pass between the GP-19 logic board connector P2 and the Z19 logic board via a 40 pin ribbon cable which plugs into the socket on the Z19 logic board formerly occupied by the Z80 microprocessor. This allows the GP-19 logic board to utilize the existing terminal I/O logic and RAM devices on the Z19 logic board. The Z80 microprocessor, which is moved to the GP-19 logic board, does not utilize the existing program ROM on the Z19 logic board; this is replaced entirely by three program ROM's (U26, U25, and U24) on the GP-19 logic board. All video generation logic on the Z19 logic board is supplanted by the GP-19 logic board.

In the case of Z19 (or Z19-HN, H19, H19A, or WH-19) terminals, a single cable connects the GP-19 logic board to the terminal power supply and video circuit board. This existing cable mates with connector P6 (or P5) on the GP-19 logic board. This cable supplies all power to the GP-19 and also routes the HORZ, VERT, and VIDEO signals generated by the GP-19 logic board to the terminal video circuit board. A second cable, supplied with the GP-19, connects P8 (or P7) on the GP-19 logic board to P401 on the Z19 logic board. This cable serves only to supply power to the Z19 logic board.

In the case of Z19-CN terminals there are separate cables from the terminal power supply and the terminal video circuit board. These existing cables both mate, side by side, with connector P5 on the GP-19 logic board. However, in the case of Z19-CN terminals, the unregulated 5 volt logic supply for the GP-19 logic board is not supplied thru this connector. Instead an auxiliary supply, mounted in the right rear corner of the terminal, connects via a new cable to P3 on the GP-19 logic board. Another cable, supplied with the GP-19, connects P7 on the GP-19 logic board to P401 on the Z19 logic board. This cable serves only to route power from the terminal power supply to the Z19 logic board.

## CIRCUIT DESCRIPTION

### 6.2 POWER SUPPLY

All circuitry on the Z19 logic board operates from 5 volt power regulated by two 7805 type three-terminal regulators designated VR1 and VR2. The outputs of these two regulators, marked VCC1 and VCC2 on the schematics, are not interconnected at any point on the board. The loads are distributed so as to approximately equalize the thermal dissipation of VR1 and VR2. Unregulated input power for VR1 and VR2 enters the board at either connector P3 or connector P6, depending on the model of terminal in which the GP-19 is installed (see Section 6.1). No +15 or -12 (-16) power is used on the GP-19 logic board. However, these supplies are made available at the auxiliary I/O connector for use by an option board.

### 6.3 CLOCK LOGIC

Two separate asynchronous clock sources are used on the GP-19 logic board. One of these is Z80 CLK, which is generated by the Z19 logic board and enters the GP-19 logic board at connector P2 pin 6. This signal is buffered and inverted by U17-C and applied to pin 6 of U27, the Z80 microprocessor clock input. It is also buffered and inverted by U17-B and then supplied to the CRT controller chip select generation logic (see paragraph 6.5.1).

The other clock source is DOT CLK, the fundamental clock for all video generation logic. The period of one cycle of dot clock is the time separating two adjacent dot positions on a horizontal scan line. Since the number of dot positions on a horizontal line depends on the display mode of the terminal, the actual frequency of DOT CLK is determined by the current display mode. In all 80 column text modes DOT CLK is 12.292 MHZ. In all 132 column text modes DOT CLK is 20.275 MHZ. And in all graphics modes DOT CLK is 10.644 MHZ.

#### 6.3.1 Oscillators

The three possible DOT CLK frequencies are generated by three separate crystal oscillators. The 12.292 MHZ oscillator, which is identical in design to the other two except for crystal frequency, uses inverter U57-D as its active gain stage. The series combination of C2 and C7 provides the proper load capacitance for crystal Y2. The two resistors serve to DC bias U57-D into its linear region. The 0.1 uF capacitor serves as a bypass to prevent AC feedback thru the resistors. The output of the oscillator is buffered and inverted by U57-E which supplies the 12.292 MHZ signal to input D0 (pin 4) of multiplexer U51. The 10.644 MHZ and 20.275 MHZ oscillators function in the same manner and their buffered outputs are applied to inputs D1 and D2

## CIRCUIT DESCRIPTION

(respectively) of multiplexer U51. Multiplexer U51 selects the proper clock oscillator as determined by multiplexer address inputs A and B (pins 11 and 10). Signals GRP/CHR H and 132/80 CHR H, which drive address inputs A and B, are supplied by outputs Q0 and Q1 of octal D flip-flop U5 (see paragraph 6.4.1)). Signal GRP/CHR H determines whether the terminal is in a graphics mode (signal high) or a text mode (signal low). Signal 132/80 CHR H determines whether the text mode screen format is 132 columns wide (signal high) or 80 columns wide (signal low). It is illegal for both these signals to be high simultaneously. The selected oscillator signal, DOT CLK H, is available at multiplexer U51 output Y (pin 5).

### 6.3.2 Timing Signal Generation

Signal DOT CLK H drives an eight state switch-tail ring counter made up of the four flip-flops in package U44. The four flip-flops each have a non-inverted and an inverted output, giving the counter a total of eight outputs which are labeled T0, T1, T2, T3, T0\*, T1\*, T2\*, and T3\* (the "\*" indicates an inverted signal). Each of these outputs has a frequency of (DOT CLK)/8 and a phase which differs from the preceding output by 45 degrees. All the timing signals shown in the Timing Diagram on page H-8 are simple logical combinations of these outputs. Gate U46-C will reset the counter to a legal state in case the counter becomes trapped in a sequence of illegal states after power up. The number of legal states in this counter (eight) is the same as the number of horizontal dots in a character cell. Thus CHR CLK H, the signal whose rising edge corresponds to the beginning of each new character cell during a horizontal sweep, is simply a buffered version of T0.

## 6.4 CPU AND SYSTEM CONTROL LOGIC

The GP-19 uses the same Z80 microprocessor that was used by the Z-19 logic board. This device is relocated to position U27 on the GP-19 logic board as part of the installation procedure. The Z80 is driven by a 2.048 MHz clock that is supplied by the Z19 logic board (see Section 6.3). The Z80 fetches and executes instructions permanently stored in ROM's U26, U25, and U24; each of which contains 4096 bytes of stored program. The Z80 can also fetch instructions from an additional ROM located on a card plugged into the auxiliary I/O connector.

The Z80 can request that the external circuitry perform four different kinds of operations: memory read, memory write, I/O read and I/O write. Memory read operations are requested by asserting signals CPU RD L (low) and CPU MREQ L (low) while outputting the address of the requested memory location on address bus lines CPU A0-15H. The external circuitry responds by making the memory data available on data bus lines



## CIRCUIT DESCRIPTION

CPU D0-7H. In the case of a memory write the Z80 supplies the address on CPU A0-15H, the data to be written on CPU D0-7H, and asserts CPU WR L (low) and CPU MREQ L (low). I/O operations are similar except CPU IOREQ L is asserted low to request the operation and only lines CPU A0-7H are used to specify the address.

The external circuitry can also request the Z80 to perform maskable or non-maskable program interrupts. Maskable interrupts are requested by asserting Z80 input INT\* (U27 pin 16 low), which is done by the output of AND gate U48-A. This can be the result of a request from the Z19 logic board (P2 pin 16 low) or a request from the auxiliary I/O card (EXT INT L low). Non-maskable interrupts are requested by asserting Z80 input NMI\* (U27 pin 17 low). This is done by the vertical retrace interrupt logic explained in paragraph 6.5.3.

### 6.4.1 Memory And I/O Port Mapping

U23 is a 3-to-8 decoder which generates the chip enable signals for the program ROMs. Enable inputs G1 (pin 6), G2A (pin 4), and G2B (pin 5) are driven by signals CPU MREQ H, CPU A15H, and CPU RD L (respectively) so that chip enable signals can only be generated when the Z80 is requesting memory read access to a location in the lower 32K of the 64K byte address space. U23 generates chip enable signals labeled ROM1 L, ROM2 L, ROM3 L and EXT MEM L. Signal EXT MEM L is routed to the external I/O connector for use by an additional program ROM on the card in that socket. The addresses to which each chip enable signal responds are shown in Table 6.1.

The GP-19 logic board utilizes the existing RAMs on the Z19 logic board for program variable space and input and output buffering. These RAMs are mapped into the address space as shown in Table 6.1. This mapping is accomplished by modifying the Z80 address and control signals before they are passed on to the Z19 logic board via P2. These signals are modified in the following manner: A11 and A12 are always high, A15 is connected to CPU A13H, and MREQ\* is only asserted (low) if CPU A15H is low and CPU MREQ L is low.

## CIRCUIT DESCRIPTION

### TABLE 6.1

#### GP-19 Memory Address Decoding

| A15 | A14 | A13 | A12 | Address Range (Hex) | Memory Element  | Part Location   |
|-----|-----|-----|-----|---------------------|-----------------|-----------------|
| 0   | 0   | 0   | 0   | (0000..0FFF)        | Program ROM 1   | U26             |
| 0   | 0   | 0   | 1   | (1000..1FFF)        | Program ROM 2   | U25             |
| 0   | 0   | 1   | 0   | (2000..2FFF)        | Program ROM 3   | U24             |
| 0   | 0   | 1   | 1   | (3000..3FFF)        | Program ROM 4   | External I/O    |
| 0   | 1   | 0   | X   | (4000..5FFF)        | 256 Byte RAM    | Z19 Logic Board |
| 0   | 1   | 1   | X   | (6000..7FFF)        | 2048 Byte RAM   | Z19 Logic Board |
| 1   | X   | -   | -   | (8000..FFFF)        | 16K Dynamic RAM | U28-U35         |

I/O port mapping is accomplished by the combination of gates U19-B, U18-C, and 3-to-8 decoder U16. The first four outputs of U16 (Y0-Y3) correspond to addresses already used for I/O operations on the Z19 logic board. The remaining four outputs (Y4-Y7) are asserted (low) whenever the Z80 performs an I/O operation on the CRT controller registers (Y4), octal flip-flop U5 (Y5), the switch register SW1 (Y6), or the card installed in the Auxiliary I/O connector (Y7). The addresses to which these outputs respond are shown in Table 6.2.

### TABLE 6.2

#### GP-19 I/O Address Decoding

| A7 | A6 | A5 | A4 | A3 | Address Range | Output of U16 | Function                 |
|----|----|----|----|----|---------------|---------------|--------------------------|
| 0  | 1  | 0  | 0  | 0  | (40-47)       | Y0            | Used by Z19 logic board  |
| 0  | 1  | 0  | 0  | 1  | (48-4F)       | Y1            | Used by Z19 logic board  |
| 0  | 1  | 0  | 1  | 0  | (50-57)       | Y2            | Used by Z19 logic board  |
| 0  | 1  | 0  | 1  | 1  | (58-5F)       | Y3            | Used by Z19 logic board  |
| 0  | 1  | 1  | 0  | 0  | (60-67)       | Y4            | R/W 6845 CRTC U50        |
| 0  | 1  | 1  | 0  | 1  | (68-6F)       | Y5            | Write octal latch U5     |
| 0  | 1  | 1  | 1  | 0  | (70-77)       | Y6            | Read switch register SW1 |
| 0  | 1  | 1  | 1  | 1  | (78-7F)       | Y7            | Auxiliary I/O connector  |

## CIRCUIT DESCRIPTION

When U16 output Y5 (pin 10) is asserted the leading edge of this signal clocks the current contents of the CPU data bus (CPU D0-7H) into octal D register U5. Outputs Q0 and Q1 of U5, signals GRP/CHR H and 132/80 CHR H, determine the screen format of the terminal as explained in paragraphs 6.3.1, 6.7.1, and 6.7.2. Output Q2, REV SCREEN H, causes all displayed characters and graphics to appear as dark symbols on a light background (see paragraph 6.7.3). Output Q3, ALT CHR L, determines which character set in the character generator ROM is currently in use (see paragraph 6.7.1). Outputs Q4-Q7 control the LED indicators; an LED is illuminated whenever an output is low.

When U16 output Y6 (pin 9) is asserted (low) the contents of switch register SW1 are enabled to data bus CPU D0-7H by octal tri-state driver U14 so they can be read by the Z80 CPU. Each switch that is closed produces a low state on the corresponding data bus line.

### 6.5 CRT CONTROLLER

The GP-19 uses the same Motorola 6845 CRT controller (or Hitachi 46505) that was used by the Z-19 logic board. This device is relocated to position U50 on the GP-19 logic board as part of the installation procedure. The 6845 is a fully programmable device with 19 internal registers whose contents are written by the Z80. The contents of these registers determine the format and timing of the signals generated by the 6845.

#### 6.5.1 Read/write Logic

The 6845 has eight bidirectional data lines (pins 33-26) which are connected to the CPU data bus (CPU D0-7H). It also has four control inputs, labeled RS, R/W, CS\*, and E, which are used to control read and write operations between the external data bus and the internal registers. Input RS (pin 24) determines whether the operation will be performed on one of the data registers (signal high) or the address register (signal low). The contents of the address register select which data register is available for I/O operations. Input R/W (pin 22) on the 6845 determines whether the internal register is to be read (signal high) or written (signal low). These inputs are connected to CPU A0H and CPU A1H (respectively) so that the the low order two bits of the I/O address determine the nature of the operation performed. Input E (pin 23) serves as a strobe to initiate a read or write operation to the internal registers of the 6845. Input CS\* is a chip enable signal; when this signal is asserted (low) the 6845 is enabled to perform read or write operations whenever the E input is strobed.

## CIRCUIT DESCRIPTION

The CS\* and E inputs to the 6845 must be asserted in the proper sequence to perform read and write operations to or from the internal registers. The CS\* line must be asserted (low) before the E pulse begins and must remain asserted until after the E pulse ends. When the CRT controller is addressed by the Z80, output Y4 of I/O address decoder U16 goes low and clears flip-flop U15-A via its direct clear input (pin 1). This causes the Q output of the flip-flop (U15-A pin 5) to go low and thus asserts CS\* at the 6845 (pin 25). The Q\* output of the same flip-flop (U15-A pin 6) goes high and thus drives the D input of flip-flop U15-B (pin 12) to a high level. At the same time, the signal from U16 output Y4 (pin 11) is inverted by U17-E and drives the direct clear input of flip-flop U15-B (pin 13) high so that the flip-flop is no longer held in the cleared state. The next positive transition of CPU clock signal Z80 CLK (buffered by gate U17-B) clocks flip-flops U15-A and U15-B. Since the D input of U15-B is high, this flip-flop sets and its Q output (pin 9) goes high and asserts input E at the 6845 (pin 23). This arrangement assures that E will be asserted about one Z80 CLK period after CS\* was asserted. When the Z80 I/O operation is complete, output Y4 of decoder U16 goes high and, after inversion by U17-E, drives the direct clear input of flip-flop U15-B (pin 13) low. The flip-flop is thus cleared and its Q output (pin 9) goes low so that the E input to the 6845 (pin 23) is no longer asserted. The clear input to flip-flop U15-A also goes high at the same time, but the Q output remains low until the next positive transition of the CPU clock. This clock transition sets U15-A, because its D input (pin 2) is being held high by output Y4 of decoder U16, causing its Q output (pin 5) to go high so that CS\* is no longer asserted at the 6845 input (pin 24). Thus CS\* ends about one Z80 CLK period after E goes low.

### 6.5.2 Display Control Signals

The 6845 produces the following signals that are used to drive the display generation circuitry:

1. CRT A0-13H These signals, which are output on 6845 pins 4-17, are the fourteen bit address of the next character in memory to be displayed. This address advances after every negative transition of CHR CLK L. The pattern of addresses generated is determined by the values in the 6845's internal registers.
2. CRT R0-3H These signals, which are output on 6845 pins 38-35, are the row address for the character generator ROM. This address determines which row of dots from the ten row character cell should be displayed on the current horizontal scan line.
3. CRT HSYNC This signal is output from the 6845 at pin 39 and buffered by gate U18-A before leaving the board. This signal determines the frequency and phase of the horizontal sweep

## CIRCUIT DESCRIPTION

generated on the video circuit board.

4. CRT VSYNC This signal is output from the 6845 at pin 40 and is inverted and buffered by inverter U7-D before leaving the board. This signal determines the frequency and phase of the vertical sweep generated on the video circuit board. It also drives the circuitry which generates the vertical retrace interrupt (see paragraph 6.5.3).
5. CRT DSEN H This signal, which is output from the 6845 at pin 18, is asserted (high) whenever the CRT beam position corresponds to an unblanked region of the display.
6. CRT CURS H This signal, which is output from the 6845 at pin 19, is asserted (high) whenever the CRT beam is positioned to a region of the screen where a cursor should be displayed.

### 6.5.3 Vertical Retrace Interrupt

Flip-flops U6-A and U6-B generate the signal NMI L which is used to interrupt the Z80 at the beginning of every vertical blanking period. This interrupt allows the Z80 to modify the 6845 internal registers and/or the display memory at a time when the display is certain to be blanked. The D input of flip-flop U6-B (pin 12) is driven by address bus bit CPU A2H. The clock input of U6-B (pin 11) is driven by the complemented CRT controller I/O select signal that originates at I/O decoder U16 and is inverted by U17-E. The program being executed by the Z80 enables the vertical retrace interrupt by addressing the CRT controller with bit two of the I/O address high, thus setting flip-flop U6-B. When flip-flop U6-B sets, its Q output goes high so that the direct clear input of flip-flop U6-A (pin 1) is no longer held in the low (asserted) state. Signal CRT VSYNC drives the clock input of flip-flop U6-A (pin 3) and the D input of the same flip-flop (pin 2) is permanently high. The next positive transition of CRT VSYNC sets flip-flop U6-A and causes output Q\* to go low, thus asserting signal NMI L at the Z80 (U27 pin 17). The Z80 program immediately branches to the vertical retrace interrupt service routine, a portion of which addresses the CRT controller with bit two of the I/O address high so as to clear U6-B (and thus U6-A) and clear the interrupt request.

## CIRCUIT DESCRIPTION

### 6.6 MEMORY AND MEMORY ARBITRATION LOGIC

The display memory operates synchronously with the other elements of the display generation logic. As was explained in paragraph 6.5.2, each negative transition of CHR CLK L causes the 6845 CRT controller to make a new display address available on bus CRT A0-13H. The display memory location specified by this address is then read by the actions of signals RAS L, CAD/RAD H, CAS L and MEM CLK H (as explained in paragraph 6.6.1) and the contents made available for display by the character and dot generation logic (see Section 6.7). However, the refresh logic and the Z80 CPU can also request access to the display memory, and a request from either of these sources will preempt the next memory cycle. If the memory cycle is preempted, then the contents of the memory location specified by the CRT controller are not made available to the character and dot generation logic and nothing is displayed at the corresponding position on the screen during that horizontal scan. In the case of the Z80 CPU accessing the display memory the situation is further complicated by the fact that the Z80 operates from a separate, asynchronous, clock source than does the display memory and display logic. Additional logic is thus necessary to synchronize Z80 CPU access to the display memory.

The display memory is comprised of eight 16K x 1 bit dynamic RAMs. The supporting circuitry necessary to implement the actions described in the previous paragraph includes address multiplexers, arbitration and synchronization logic, and refresh cycle generation logic. The operation of each of these elements is described in the following paragraphs.

#### 6.6.1 Dynamic Memory Operation

The type 4517 dynamic RAMs each have only seven address input pins. Since the required address is 14 bits, it must be clocked into the RAMs in two seven bit pieces. The sequence of events involved in a read operation is as follows:

When the least significant seven bits of the address become available at 4517 inputs A0-A6 (pins 5-7, 12, 11, 10, and 13), clock RAS L is asserted (low) at pin 4 of each 4517. The negative transition of this clock causes the first seven bits of the address to be latched into the internal row address buffers in the 4517s. This portion of the address is then used to select, read, and refresh one row (128 locations) within each RAM. Shortly after RAS L is asserted, the signal CAD/RAD H goes high and causes the most significant seven bits of the memory address to be made available at the 4517 address inputs (see paragraph 6.6.3). When this portion of the address has stabilized the signal CAS L is asserted at pin 15 of each 4517. This negative transition causes the most significant seven bits of the memory address to be latched into the

## CIRCUIT DESCRIPTION

internal column address buffers in the 4517s. This portion of the address is then used to select the contents of one of the 128 locations in the row read by the action of RAS L and make it available at the Q output (pin 14) of each RAM. The selected value remains available so long as CAS L is asserted. The Q outputs of the eight RAMs are connected to the eight D inputs of octal flip-flop U11. Since the clock input of U11 (pin 11) is driven by signal MEM CLK H, the next positive transition of MEM CLK H causes the byte resulting from the RAM read operation to be stored in U11. The relationship between RAS L, CAD/RAD H, CAS L and MEM CLK H can be seen in the Timing Diagram on page H-8.

A display memory write operation involves only one additional clock signal. The timing of the signals RAS L, CAD/RAD H, and CAS L is the same as was described for read operations. If a write operation is to take place, signal MEM WRT L is asserted (low) at the same time as signal RAS L. MEM WRT L drives input pin 3 of each of the 4517 RAMs low, thus signaling that the memory cycle just begun will be a write cycle. Signal MEM WRT L also drives low the enable inputs of octal bus driver U13 (pins 1 and 19), thus enabling the contents of bus CPU D0-7H (the Z80 data bus) to bus MEM D0-7H (the dynamic RAM data bus). This makes the contents of the Z80 data bus available at the D input pin of each 4517 (pin 2). Because MEM WRT L is asserted the 4517s do not enable their Q outputs when CAS L is asserted, instead they write the value of the D inputs (pin 2 of each RAM) into the location that is specified by the Row and Column addresses. Signal MEM WRT L is generated by gates U45-C and U46-B. Gate U45-C derives the timing of MEM WRT L from basic timing signals T0 and T2; the output of this gate drives one input of three-input NAND gate U46-B (pin 5). The other two inputs of U46-B are driven by signals CPU RD L (pin 4) and PRC GRANT H (pin 3), so that MEM WRT L will only be asserted if the Z80 is requesting a write operation and the Z80 has been granted access to the display memory.

### 6.6.2 Memory Access Arbitration And Synchronization

The state of flip-flops U54-A and U54-B determines whether a given display memory cycle will be used by the Z80 CPU, the memory refresh logic, or the display generation logic. If U54-A is clear then its Q\* output (pin 6) is high and asserting signal PRC GRANT H, indicating the current memory cycle has been granted to the Z80 CPU. If U54-B is clear then its Q\* output (pin 8) is high and asserting signal RFS GRANT H, indicating the current memory cycle has been granted to the memory refresh logic. If neither of these flip-flops is clear then the current memory cycle is implicitly granted to the display generation logic. The case where both flip-flops are clear simultaneously is not valid.

## CIRCUIT DESCRIPTION

Flip-flops U54-A and U54-B are both clocked by the positive transition of signal CAS L. This positive transition causes the two flip-flops to be set or cleared according to the state of their D inputs. The D input of flip-flop U54-B (pin 12) is driven by signal RFS REQ L, the Q\* output of flip-flop U52-A (pin 6). Flip-flop U52-A will be set whenever it is time to perform another memory refresh cycle (see paragraph 6.6.4). If flip-flop U52-A is set, then the D input of U54-B is low and the next positive transition of CAS L causes flip-flop U54-B to be cleared and thereby asserts RFS GRANT H. The D input of flip-flop U54-A (pin 2) is driven by the output of NAND gate U55-A (pin 3). One input of NAND gate U55-A (pin 2) is driven by the signal RFS REQ L from the Q\* output of flip-flop U52-A (pin 6). The other input of NAND gate U55-A is driven by the Q output of flip-flop U53-A (pin 5), which is set whenever the Z80 CPU is currently waiting to be granted a display memory cycle. Thus the output of NAND gate U55-A (pin 3) only drives the D input of flip-flop U54-A low if the Z80 CPU is requesting access to the display memory and the memory refresh logic is not requesting a display memory cycle. This assures that requests for memory refresh cycles always have priority over Z80 CPU requests for display memory cycles.

Signal PRC REQ H is asserted (high) by the output of inverter U56-B (pin 4), as a result of the combined action of gates U46-A, U47-D, U48-D, U56-A and U56-C, and indicates that the Z80 CPU is requesting access to the display memory. PRC REQ H is asserted whenever one of the following two combinations of conditions is met:

1. The Z80 CPU is attempting to read or write data to or from an address in the range of 8000-FFFF (Hex) and the hardware is currently in character mode (GRP/CHR H low).
2. The Z80 CPU is attempting to read or write data to or from an address in the range of C000-FFFF (Hex) and the hardware is currently in graphics mode (GRP/CHR H high).

Signal PRC REQ H drives the clock input of flip-flop U53-A (pin 3). The D input of flip-flop U53-A (pin 2) is always high. Thus when PRC REQ H becomes asserted (makes a positive transition) flip-flop U53-A is set as a result. When flip-flop U53-A becomes set its Q\* output (pin 6) goes low and asserts signal MEM WAIT L, which in turn drives one input of AND gate U48-B (pin 5) low. As a result, the output of gate U48-B goes low and asserts the WAIT\* input of the Z80 CPU (pin 24). Asserting the WAIT\* input of the Z80 causes the Z80 to execute internal "wait" states after completing internal clock state T2. The Z80 will not continue on to execute internal clock state T3 until such time as the WAIT\* input is no longer asserted. This mechanism allows the Z80 CPU to "stall" until such time as the access to display memory that it is requesting can be granted.



## CIRCUIT DESCRIPTION

When a display memory cycle is granted to the Z80 CPU, flip-flop U54-A is cleared and thus asserts signal PRC GRANT H (as was explained in the preceding paragraphs). PRC GRANT H is ANDed with signal WAIT CLR H by NAND gate U47-B and the output of this gate (pin 6, asserted when low) drives one input of gate U48-C, the output of which in turn drives the direct clear input of flip-flop U53-A (pin 1). The result is that if the current display memory cycle has been granted to the Z80 CPU (PRC GRANT H asserted), then at the time in the cycle when WAIT CLR H is asserted (see the Timing Diagram on page H-8) flip-flop U53-A will be cleared. This causes signal MEM WAIT L to go high (not asserted) and release the Z80 CPU from its wait state.

If the current display memory cycle is a read operation by the Z80 CPU then U12 serves both as an octal transparent latch to retain the byte read from memory and as an octal tri-state driver to place this information on the Z80 CPU data bus CPU D0-7H. The output of gate U22-A (pin 3) asserts the output driver control input of U12 (pin 1), thus enabling the outputs of U12 to bus CPU D0-7H, whenever the Z80 CPU is requesting read access to the display memory. The latch enable input of U12 (pin 11) is driven by signal MEM LCH H, which is the Q output of flip-flop U53-B. Whenever the flip-flop is set (MEM LCH H asserted) U12 is in the transparent state, i.e., the data outputs follow the data inputs. When flip-flop U53-B is cleared, MEM LCH H goes low and the data outputs of U12 become latched in the state they were in at the time of the transition. The clock input of flip-flop U53-B (pin 11) is driven by signal PRC GRANT H and the D input (pin 12) is always high; thus the flip-flop is set and MEM LCH H goes high whenever PRC GRANT H becomes asserted. Therefore, the contents of memory data bus MEM D0-7H are enabled to Z80 CPU data bus CPU D0-7H as soon as a display memory cycle is granted to the Z80 CPU. When the memory cycle progresses to the point where the read memory data is stable, flip-flop U53-B is direct-cleared by signal MEM CLK H acting thru gate U47-C. This causes MEM LCH H to go low so that the memory data is latched in the outputs of U12 and remains available to the Z80 even after the display memory starts a new cycle.

### 6.6.3 Address Generation

The seven 8-to-1 multiplexers, U36-U42, drive the seven address lines of the 4517 dynamic RAMs. These multiplexers serve two different functions:

1. They select the source of the memory address for the current memory cycle. The source may be the display address output by the 6845 CRT controller, a combination of the display address and character row address output by the 6845 CRT controller, the Z80 CPU, or the refresh logic.

## CIRCUIT DESCRIPTION

2. They select either the most significant seven bits or the least significant seven bits of the 14 bit display memory address, depending on which half of the memory cycle is currently in progress.

The least significant address input of each multiplexer (input A, pin 11) is driven by signal CAD/RAD H which is output by gate U43-D (pin 11). This signal is high during the portion of the memory cycle when the column address is required, and low during the portion when the row address is required (see paragraph 6.6.1). The other two address inputs B and C (pins 10 and 9) are driven by the outputs of gates U55-B and U55-D, which in conjunction with gate U55-C, decode the source of the address. The four possibilities are as follows:

1. The memory cycle has been granted to the display generation logic and the hardware is in character mode. When CAD/RAD H is low the multiplexer D0 inputs will be selected and the resulting address will be CRT A0-6H. When CAD/RAD H is high the multiplexer D1 inputs will be selected and the resulting address will be CRT A7-13H.
2. The memory cycle has been granted to the display generation logic and the hardware is in graphics mode. When CAD/RAD H is low the multiplexer D2 inputs will be selected and the resulting address will be CRT A0-5H, CRT ROH. When CAD/RAD H is high the multiplexer D3 inputs will be selected and the resulting address will be CRT A6-12H.
3. The memory cycle has been granted to the Z80 CPU. When CAD/RAD H is low the multiplexer D4 inputs will be selected and the resulting address will be CPU A0-6H. When CAD/RAD H is high the multiplexer D5 inputs will be selected and the resulting address will be CPU A7-13H.
4. The memory cycle has been granted to the memory refresh logic. When CAD/RAD H is low the multiplexer D6 inputs will be selected and the resulting address will be RFS A0-6H. No column address is required for refresh operations.

### 6.6.4 Refresh Logic

In order to guarantee that the 4517 dynamic RAMs will retain data, a memory cycle at each of the 128 possible row addresses must be performed every two milliseconds or less. The refresh logic insures that this criteria will be met by performing memory refresh cycles during each horizontal retrace period.

## CIRCUIT DESCRIPTION

The clock input of flip-flop U52-A (pin 3) is driven by signal CHR CLK H and the D input (pin 2) is driven by signal CRT HSYNC. Thus the first positive transition of CHR CLK H that occurs after CRT HSYNC becomes high (the beginning of horizontal retrace) causes flip-flop U52-A to be set, and the flip flop remains set until the first positive transition of CHR CLK H that occurs after CRT HSYNC goes low. When flip-flop U52-A becomes set, its Q\* output goes low and asserts signal RFS REQ L. As was explained in paragraph 6.6.2, signal RFS REQ L requests that display memory cycles be granted to the refresh logic; since this is the highest priority request it is always granted at the beginning of the next memory cycle. The Q output of flip-flop U52-A drives high one input of NAND gate U45-B (pin 4), the other input of gate U45-B (pin 5) is driven by signal MEM CLK H. The output of gate U45-B (pin 6), which goes low each time MEM CLK H goes high during a horizontal retrace period, drives the inverted clock input of the asynchronous counter consisting of U49-A and U49-B. Since MEM CLK H goes high at the end of each memory cycle, this means that the refresh address supplied by U49-A and U49-B (RFS A0-6H) is incremented following every memory cycle that was granted to the memory refresh logic.

### 6.7 CHARACTER AND DOT GENERATION

As was explained in paragraph 6.6.1, each byte that is read from display memory is latched into octal D flip-flop U11 by the positive transition of signal MEM CLK H. If the memory cycle that produced the byte was granted to the Z80 CPU or the refresh logic, rather than the display logic, then the byte is not valid for display and U11 is immediately direct-cleared by the combined action of gates U18-D, U19-C, and inverter U7-A.

The outputs of octal D flip-flop U11, with the exception of Q5 (pin 15), are connected directly to the inputs of octal D flip-flop U10. U11 and U10 form a "pipeline" which makes a byte from memory available for display on the second cycle of CHR CLK H after the transition of CHR CLK H that clocked the 6845 CRT controller and thus generated the display address. This pipeline compensates for the fact that the combined action of the 6845 CRT controller and the 4517 dynamic RAMs is to slow to read a byte from display memory in a single cycle of CHR CLK H when the hardware is operating in the 132 character line mode.

If the hardware is in character mode rather than graphics mode, then whenever U11 is direct-cleared (because the byte was not valid for display) input D5 of U10 (pin 14) is forced high by the combined action of gates U22-C and U19-A. This causes the byte latched into U10 by the action of CHR CLK H to have a value of 20 (Hex), the code for an ASCII blank, so that a blank will ultimately be displayed at the corresponding position on the screen during that horizontal scan. If the hardware is in graphics mode, and U11 has been direct-cleared because the byte was

## CIRCUIT DESCRIPTION

not valid for display, then input D5 of U10 is not forced high and the byte latched into U10 has a value of 00 (Hex). This causes no graphics dots to be turned on at the corresponding position on the screen during that horizontal scan line.

### 6.7.1 Character Generation

Character generator U21 is a 4096 x 8 ROM which contains the dot matrix representations of the characters in the terminal's character sets. The most significant address bit, A11 (pin 21), is driven by signal ALT CHR L; this signal selects which of the two character sets stored in U21 is currently in use. Signal ALT CHR L is driven by output Q3 of octal D flip-flop U5, the contents of which are loaded by the Z80 CPU (see paragraph 6.4.1).

Address inputs A4-A10 of character generator ROM U21 are driven by outputs Q0-Q6 of U10, the least significant seven bits of the byte read from the display memory. These inputs select the particular character within the ROM which is to be displayed. Address inputs A0-A3 of U21 are driven by signals CRT R0-3H, the character row address generated by the 6845 CRT controller. These inputs determine which of the ten possible horizontal rows of dots that make up the character should be output on a particular scan line. The character row address only changes between horizontal scan lines, it is always the same for all characters displayed on a given horizontal scan.

Outputs D0-D7 of character generator U21 contain the values of the eight dots that make up the selected horizontal row of the selected character. Output D0 controls the right-most dot in the selected row of the character cell (0=no dot, 1=dot) and D7 controls the left most dot. These outputs drive the eight parallel data inputs, labelled A-H, of shift register U20.

Eight bit shift register U20 is clocked by signal DOT CLK H, which has a period equal to the time separating two adjacent dot positions on a horizontal scan line (see Section 6.3). The shift/load control input of U20 (pin 15) is driven by signal SHIFT/LD H. When this signal is low at the time a positive transition of DOT CLK H occurs, the shift register is parallel loaded with the eight bits being output by character generator ROM U21. In this case the data on input H (pin 14) becomes available at output QH (pin 13) immediately after the positive transition of DOT CLK H. As can be seen in the Timing Diagram on page H-8, signal SHIFT/LD H goes low for one period of signal DOT CLK H immediately preceding the positive transition of signal CHR CLK H. This causes the positive transition of DOT CLK H that occurs simultaneously with the positive transition of CHR CLK H to parallel load shift register U20 with the eight bit output of character generator U21. These eight bits represent a row of eight horizontal dots generated from

## CIRCUIT DESCRIPTION

the display memory byte that was loaded into U10 by the previous positive transition of CHR CLK H. When signal SHIFT/LD H is high at the time a positive transition of DOT CLK H occurs the contents of the register are shifted one position. This shifts the data latched in position QG to position QH (so it is now available at the output), the data in position QF to position QG, etc. The result is that after each positive transition of DOT CLK H the value for the dot at the next position on the horizontal scan line becomes available at output QH of U20.

The direct clear input of shift register U20 (pin 9) is driven by signal GRP/CHR L. If the hardware is in character mode then signal GRP/CHR L is high and the shift register is not held clear. However, if the hardware is in graphics mode then shift register U20 is held continuously clear and output QH is always low.

### 6.7.2 Graphics Dot Generation

When the hardware is in graphics mode the character generator ROM is not used. Instead, each of the eight bits that make up the byte read from display memory control a dot on the horizontal scan line. Each byte read from display memory thus specifies an eight dot segment of the horizontal scan line. The eight outputs of U10 drive the eight inputs of shift register U9, with output Q0 of U10 (pin 2) connected to input H of U9 (pin 14) and output Q7 of U10 (pin 19) connected to input A of U9 (pin 2). Output Q0 of U10 controls the left-most dot in the segment of the horizontal scan line (1=dot, 0=no dot) and output Q7 of U10 controls the right-most dot in the segment. .P; Shift register U9 operates in the same manner as was explained above for shift register U20; it is loaded and shifted by the same signals. However, its direct clear input is driven by GRP/CHR H, the inversion of the signal that drives the direct clear input of shift register U20. This means that when the hardware is in character mode, shift register U9 is held clear and its QH output is always low.

### 6.7.3 Blanking, Reverse Video, And Reverse Screen

The QH outputs of shift registers U20 and U9 are ORed together by gate U22-D. The output of gate U22-D, which is the dot stream produced by the currently enabled shift register, drives input D0 of hex flip-flop U1 (pin 3). U1 is clocked by signal DOT CLK H so that the dots that reappear at output Q0 are synchronized with the positive transitions of DOT CLK H; thus any dot timing jitter accumulated in the dot stream up to this point is eliminated. The dot stream then undergoes the following additional gating:

## CIRCUIT DESCRIPTION

1. If the hardware is in character mode, and the most significant bit of the byte that generated the character is set, then the character is to be displayed as reverse video and the value of the dot is inverted by EXOR gate U2-D. If output Q2 of U5 is high then the hardware is in reverse screen mode and the value of the dot is inverted by EXOR gate U2-D. If both these conditions are true the the value of the dot is not inverted by EXOR gate U2-D.
2. If the 6845 CRT controller is requesting that a cursor be displayed at the current position on the horizontal scan line (signal CRT CURS H asserted) then the value of the dot is inverted by EXOR gate U2-C. Note that this inversion is in addition to any inversion performed by U2-D.
3. If the dot to be displayed is in a position that the 6845 CRT controller regards as part of the blanked area of the screen (signal CRT DSEN H low) then the value of the dot is forced to the "no dot" state by NAND gate U45-A.

The output of gate U45-A drives input D4 of hex flip-flop U1. The dot stream reappears at output Q4, once again synchronized with the the positive transitions of signal DOT CLK H so as to eliminate any dot timing jitter contributed by gates U2-D, U2-C, and U45-A. Output Q4 of U1 drives the inputs of inverters U7-C and U7-E, the outputs of which are tied in parallel and pulled up by 2.2K resistor R7. These outputs provide the video signal which drives the cable to the terminal video circuit board.

Octal flip-flop U8, as well as the remaining sections of flip-flop U1, are used to delay the signals that control gates U2-D, U2-C, and U45-A so that they are properly synchronized in time with the dot stream available at output Q0 of U1. These delays are necessary because the actions of U11, U10, U20 (or U9) and U1 impose a delay of three character times plus one dot time between the time the 6845 CRT controller is clocked and the time the first resulting dot value becomes available at output Q0 of U1.

The cursor and blanking control signals produced by the 6845 (CRT CURS H and CRT DSEN H) are available about three character times earlier than the dot values they ultimately modify. Signal CRT CURS H drives input D1 of octal flip-flop U8. Output Q1 drives input D2 and output Q2 in turn drives input D3. Since U8 is clocked by signal CHR CLK H the result is that signal CRT CURS H arrives at output Q3 after being delayed three character times. Output Q3 of U8 then drives input D2 of U1 so that the signal arrives at output Q2 of U1 delayed an additional one dot time, making it exactly coincident with the dots arriving at output Q0 of U1. Signal CRT DSEN H is handled in precisely the same manner, it enters U8 at input D4 and ultimately reappears at output Q3 of U1 having been delayed a total of three character times and one dot time.

## CIRCUIT DESCRIPTION

The per-character reverse video bit at output Q7 of U10 also arrives too soon, in this case about one character time. Output Q7 of U10 is ANDed with signal GRP/CHR L by gate U19-D; the output of this gate is thus only active when the hardware is in character mode rather than graphics mode. The output of gate U19-D (pin 11) is EXORed with signal REV SCREEN H by EXOR gate U2-B. The output of gate U2-B will thus be high if the current character is flagged to be displayed in reverse video, or if the terminal is in reverse screen mode, but not if both conditions are true. The output of U2-B drives input D0 of octal flip-flop U8 (pin 3); the signal reappears at output Q0 after having been delayed until the next leading edge of CHR CLK H. Output Q0 of U8 then drives input D1 of U1 so that the signal arrives at output Q1 of U1 delayed an additional one dot time, making it exactly coincident with the dots arriving at output Q1 of U1.

## APPENDIX A

### ESCAPE SEQUENCES AND CONTROL CHARACTERS

#### A.1 CONTROL CHARACTERS

Receipt of the following control codes while the terminal is in a text mode causes the specified action. For behavior in Tektronix mode, see paragraph 4.6.5.

|     |  |
|-----|--|
| ENQ | Returns 'who are you' message if auxiliary board is installed  |
| BEL | Sounds 1000 HZ tone for 200 MS   |
| BS  | Moves cursor backward one character (see paragraph 4.12.5)   |
| HT  | Moves cursor to the next horizontal tab stop or to the end of the line if no more tab stops remain                           |
| LF  | Moves cursor down one row, scrolling if necessary. If Auto-CR-on-LF is set the cursor is placed at the start of the new line |
| VT  | Moves cursor down one row, scrolling if necessary. No other operations occur   |
| FF  | Moves cursor down one row (same as VT)   |
| CR  | Moves cursor to beginning of line. If Auto-LF-on-CR is set the cursor is moved down one row as well.                         |
| S0  | Selects G1 character rendition (USASCII or graphics)<br>See Section A.13   |
| SI  | Selects G0 character rendition (USASCII or graphics)<br>See Section A.13   |



## ESCAPE SEQUENCES AND CONTROL CHARACTERS

|               |   |
|---------------|---|
| DC1<br>(XON)  | Enables transmission of characters from the terminal's output buffer if 'output synchronization' is enabled, otherwise ignored. See paragraph 4.12.13.  |
| DC3<br>(XOFF) | Disables transmission of characters from the terminal's output buffer if 'output synchronization' is enabled, otherwise ignored. See paragraph 4.12.13. |
| ESC           | Initiates start of an escape sequence (see below)   |
| GS            | Switches from text mode to Tektronix mode, Graph state  |

### A.2 ESCAPE SEQUENCE DEFINITIONS

Escape Sequences are character strings that begin with the ESC control code. When the terminal receives an ESC character it does not display subsequent characters until it receives either the final character of the sequence, a CAN control character, or an illegal character such as a SPACE. Certain control characters (particularly XON/XOFF) may be embedded in the string although this practice is not in accordance with ANSI standards. A second ESC received before the final character of the first sequence causes the first sequence to be aborted, but does not necessarily initiate a second sequence. Note that escape sequences may not contain SPACES; the separation between characters in the text below is for clarity only. An unrecognized escape sequence is ignored, as are parameters that do not correspond to executable functions.

The GP-19 terminal recognizes three distinct classes of sequences:

1. ZENITH sequences, which are recognized when the terminal is in ZENITH escape mode and are compatible with the Z19 and VT52 type terminals
2. ANSI sequences, which are recognized when the terminal is in ANSI escape mode and are compatible with the Z19 and other terminals
3. Global sequences, which are recognized without regard to what escape mode the terminal is in and are compatible only with GRAPHICS-PLUS equipped terminals.

ZENITH sequences consist of a fixed set of characters (at most four) which cause the terminal to perform a single function. ANSI and Global sequences, on the other hand, often contain variable parameters which are denoted below as 'Pn' (sometimes other labels are used, according to the context). These parameters are decimal numbers (range 0-255), and often it is possible to include several parameters (limit 15) in one sequence, using semicolons as separators. For instance, the 'select

## ESCAPE SEQUENCES AND CONTROL CHARACTERS

graphic rendition' sequence recognizes the parameters '7' (enable reverse video) and '10' (enable character graphics). The sequence (ESC [ 7;10 m) would thus enable both of those attributes. Sequences whose final character is 'h' (set), 'l' (reset) or 'm' (graphic rendition) permit multiple parameters. (Note 'reset' uses the letter 'l', not the number '1'.)

When a parameter is omitted (or has the value '0'), the default value for that sequence is used (except in Global sequences). Thus all the following are equivalent to 'ESC [ H' (the default for this sequence is '1'):

```

ESC [ ; H      ESC [ 0 ; H      ESC [ ; 0 H      ESC [ 0 ; 0 H
ESC [ 1 H      ESC [ 0 ; 1 H      ESC [ ; 1 H      ESC [ 1 ; 0 H  etc.

```

### A.3 TERMINAL MODE CONTROL

|     | <u>Function</u>  | <u>ZENITH</u> | <u>ANSI</u>                | <u>Global</u> |
|-----|--|---------------|----------------------------|---------------|
| (1) | Enter ANSI esc mode  | ESC <         |                            | ESC : 18 h    |
|     | Enter ZENITH esc mode  |               | ESC [ ? 2 h<br>ESC [ ? 2 l | ESC : 18 l    |
| (2) | EDT-compatibility on   |               |                            | ESC : 11 h    |
|     | EDT-compatibility off  |               |                            | ESC : 11 l    |
|     | Enter Tek Alpha state  | ESC FF        | ESC FF                     | ESC : 5 f     |
|     | Enter Tek Graph state  |               |                            | ESC : 6 f     |
|     | Enter Easy-Entry state   |               |                            | ESC : 7 f     |
|     | Exit Tektronix or Easy-Entry mode<br>and return to most recent text mode |               |                            | ESC : 8 f     |

Note 1: The sequence ending in 'h' is ZENITH-compatible; the sequence ending in 'l' is VT100-compatible

Note 2: EDT-compatibility applies only to ANSI mode - see paragraph 4.12.10

## ESCAPE SEQUENCES AND CONTROL CHARACTERS

### A.4 KEYPAD MODE CONTROL

| <u>Function</u>            | <u>ZENITH</u>    | <u>ANSI</u>          | <u>Global</u> |
|----------------------------|------------------|----------------------|---------------|
| Set shifted keypad         | ESC t<br>ESC x 6 | ESC [ > 6 h          |               |
| Set unshifted keypad       | ESC u<br>ESC y 6 | ESC [ > 6 l          |               |
| Enable alternate mode      | ESC =<br>ESC x 7 | ESC =<br>ESC [ > 7 h |               |
| Disable " "                | ESC ><br>ESC y 7 | ESC ><br>ESC [ > 7 l |               |
| ANSI cursor keys alternate |                  | ESC [ ? 1 h          |               |
| ANSI cursor keys normal    |                  | ESC [ ? 1 l          |               |

### A.5 ESCAPE SEQUENCE REPORTS SENT BY THE TERMINAL

| <u>In response to:</u>      | <u>The Terminal sends:</u>           |                    |
|-----------------------------|--------------------------------------|--------------------|
| ESC Z (Identify)            | ESC / K (as a VT52)                  | (ZENITH mode only) |
| ESC [ c (Attributes)        | ESC [ ? 1 ; 0 c (no options)         | (ANSI only)        |
| ESC [ 5 n (Status)          | ESC [ 0 n (terminal OK)              | (ANSI only)        |
| ESC ENQ (Tek status)        | See sections 4.6.1.3, 4.6.2.4, 4.6.3 |                    |
| (3) ESC : q (Screen format) | ESC : Pn q                           | (Global)           |
| (4) ESC n (Cursor report)   | ESC Y rowchr colchr                  | (ZENITH mode)      |
| (5) ESC [ 6 n (Cur report)  | ESC [ Pr ; Pc R                      | (ANSI mode)        |
| (5) ESC : 6 n (Cur report)  | ESC : Pr ; Pc R                      | (Global)           |
| (6) ESC : w (Window adr)    | ESC : Pn a                           | (Global)           |

Note 3: The range of Pn is 30-37 (Hex) with the three least significant bits indicating the present screen format as follows:

|                |     |
|----------------|-----|
| Tektronix mode | 001 |
| Wide Screen    | 010 |
| Long Screen    | 100 |

Note 4: The range of 'rowchr' and 'colchr' is 20-7E(Hex); columns 95-132 all return the character '~' (7E Hex).

Note 5: 'Pr' and 'Pc' are decimal numbers in the range 1-132.

Note 6: Reports the position in display memory of the home line of the screen window. The response in Tektronix mode is meaningless.

## ESCAPE SEQUENCES AND CONTROL CHARACTERS

### A.6 ESCAPE SEQUENCES SENT BY THE KEYPAD AND FUNCTION KEYS

#### Keypad Mode

| Key<br>Label | normal |       | alternate |         | EDT     |         |
|--------------|--------|-------|-----------|---------|---------|---------|
|              | ZENITH | ANSI  | ZENITH    | ANSI    | compat  |         |
| ENTER        | CR     | CR    | ESC ? M   | ESC O M | ESC O M |         |
| "."          | "."    | "."   | ESC ? n   | ESC O n | ESC O n |         |
| 0            | 0      | 0     | ESC ? p   | ESC O p | ESC O p |         |
| 1            | 1      | 1     | ESC ? q   | ESC O q | ESC O q |         |
| 2            | 2      | 2     | ESC ? r   | ESC O r | ESC O r |         |
| 3            | 3      | 3     | ESC ? s   | ESC O s | ESC O v |         |
| 4            | 4      | 4     | ESC ? t   | ESC O t | ESC O t |         |
| 5            | 5      | 5     | ESC ? u   | ESC O u | ESC O u |         |
| 6            | 6      | 6     | ESC ? v   | ESC O v | ESC O l |         |
| 7            | 7      | 7     | ESC ? w   | ESC O w | ESC O w |         |
| 8            | 8      | 8     | ESC ? x   | ESC O x | ESC O R |         |
| 9            | 9      | 9     | ESC ? y   | ESC O y | ESC O m |         |
| (7) up       | ^      | ESC A | ESC [ A   | ESC A   | ESC O A | (7)     |
| (7) down     | v      | ESC B | ESC [ B   | ESC B   | ESC O B | (7)     |
| (7) right    | >      | ESC C | ESC [ C   | ESC C   | ESC O C | (7)     |
| (7) left     | <      | ESC D | ESC [ D   | ESC D   | ESC O D | (7)     |
| Home         |        | ESC H | ESC [ H   | ESC H   | ESC [ H | ESC O u |
| IL           |        | ESC L | ESC [ L   | ESC L   | ESC [ L | ESC O s |
| DL           |        | ESC M | ESC [ M   | ESC M   | ESC [ M | ESC O S |
| DC           |        | ESC N | ESC [ P   | ESC N   | ESC [ P | ESC O y |
| (8) IC on    |        | ESC @ | ESC [4h   | ESC @   | ESC [4h | ESC O x |
| IC off       |        | ESC O | ESC [4L   | ESC O   | ESC [4L | ESC O x |
| ERASE        |        | ESC J | ESC [ J   | ESC J   | ESC [ J | ESC [ J |
| SHFT/ER      |        | ESC E | ESC [2J   | ESC E   | ESC [2J | ESC [2J |
| (9) Blue     |        | ESC P | ESC O P   | ESC P   | ESC O P | ESC O P |
| Red          |        | ESC Q | ESC O Q   | ESC Q   | ESC O Q | ESC O Q |
| White        |        | ESC R | ESC O R   | ESC R   | ESC O R | ESC O R |
| F1           |        | ESC S | ESC O S   | ESC O   | ESC O S | ESC O S |
| F2           |        | ESC T | ESC O T   | ESC T   | ESC O T | ESC O T |
| F3           |        | ESC U | ESC O U   | ESC U   | ESC O U | ESC O U |
| F4           |        | ESC V | ESC O V   | ESC V   | ESC O V | ESC O V |
| F5           |        | ESC W | ESC O W   | ESC W   | ESC O W | ESC O W |

Note 7: In ANSI escape mode the cursor keys can be set for 'normal' or 'alternate' sequences independently of the other keypad keys. The cursor keys are not affected by EDT compatibility.

Note 8: The sequences sent by the IC key toggle. The first sequence sent enables 'insert-character state', the second disables it.

Note 9: 'BLUE'-'F5' shown in unprogrammed (reset) state, independent of the Shift key. User may program other strings - see sect. 4.10.

## ESCAPE SEQUENCES AND CONTROL CHARACTERS

### A.7 CURSOR POSITION OPERATIONS

|       | <u>Function</u>  | <u>ZENITH</u>       | <u>ANSI</u>                    | <u>Global</u> |
|-------|--|---------------------|--------------------------------|---------------|
|       | Cursor home  | ESC H               | ESC [ H<br>ESC [ f             | ESC : H       |
| (10)  | Cursor Up  | ESC A               | ESC [ Pn A                     |               |
|       | Cursor Down  | ESC B               | ESC [ Pn B                     |               |
|       | Cursor Right   | ESC C               | ESC [ Pn C                     |               |
|       | Cursor Left  | ESC D               | ESC [ Pn D                     |               |
| (11)  | Index Up   | ESC I               | ESC M                          |               |
|       | Index Down   | VT                  | ESC D                          |               |
|       | New line   | CR,LF               | ESC E                          |               |
| (12)  | Position cursor  | ESC Y rowchr colchr | ESC [ Pr;Pc H<br>ESC [ Pr;Pc f | ESC : Pr;Pc H |
| (12a) | Position cursor on status line<br>and enable status line |                     |                                | ESC : Pc L    |
| (13)  | Save cursor position                                     | ESC j               | ESC 7<br>ESC [ s               |               |
| (13)  | Restore cursor position                                  | ESC k               | ESC 8<br>ESC [ u               |               |
|       | Enable status line                                       | ESC x 1             | ESC [ > 1 h                    |               |
|       | Disable status line                                      | ESC y 1             | ESC [ > 1 l                    |               |

Note 10 Attempted movement of the cursor past the first or last column (or the top or bottom row) is ignored. No scrolling occurs as a result of cursor movement. A value of '1' is assumed if Pn is omitted.

Note 11 Index operations change only the row position; moving up on the top row or down on the bottom row causes the display to scroll.

Note 12 The (row,col) values in ZENITH mode are encoded as characters in the range 20-7E (Hex). ANSI and Global sequences use decimal numbers in the range 1-132 with the value '1' assumed if no value or '0' is specified. With the exception of (ESC : Pc L) the status line must be enabled before the cursor can be positioned there.

Note 12a Before the cursor is positioned a 'save cursor' operation (see note 13) is automatically performed. The terminal can thus be subsequently restored to the prior state by ESC 8.

## ESCAPE SEQUENCES AND CONTROL CHARACTERS

Note 13 The sequences ESC 7 and ESC 8 save/restore the cursor position, the character set definitions for G0 and G1, the currently selected character set, and relative/absolute origin mode.

### A.8 SCREEN ERASURE OPERATIONS

|      | <u>Function</u>                               | <u>ZENITH</u> | <u>ANSI</u>            | <u>Global</u>        |
|------|---|---------------|------------------------|----------------------|
|      | Erase all off-screen memory                   |               |                        | ESC : E              |
|      | Erase from cursor to end of off-screen memory |               |                        | ESC : O E            |
| (15) | Erase entire screen                           | ESC E         | ESC [ 2 J              | ESC : 1 P            |
| (16) | Erase defined scrolling region                |               |                        | ESC : P<br>ESC : O P |
|      | Erase from cursor to:                         |               |                        |                      |
|      | top of screen                                 | ESC b         | ESC [ 1 J              |                      |
|      | bottom of screen                              | ESC J         | ESC [ J                |                      |
|      | beginning of line                             | ESC o         | ESC [ 0 J<br>ESC [ 1 K |                      |
|      | end of line                                   | ESC K         | ESC [ K<br>ESC [ 0 K   |                      |
|      | Erase entire line                             | ESC L         | ESC [ 2 K              |                      |

Note 15 See section 4.12.9.1 for placement of cursor following erasure.

Note 16 The cursor is positioned at the start of the scrolling region in relative origin mode, at the home position in absolute origin mode, and left unchanged in ANSI+EDT mode.

### A.9 SCREEN EDITING OPERATIONS

|      | <u>Function</u> | <u>ZENITH</u> | <u>ANSI</u> | <u>Global</u> |
|------|-----------------|---------------|-------------|---------------|
| (17) | Delete Line     | ESC M         | ESC [ Pn M  |               |
|      | Insert Line     | ESC L         | ESC [ Pn L  |               |
|      | Delete Char     | ESC N         | ESC [ Pn P  |               |
|      | Enter IC state  | ESC @         | ESC [ 4 h   |               |
|      | Leave IC state  | ESC O         | ESC [ 4 l   |               |

Note 17 ANSI mode operations with Pn greater than 1 delete chars to the RIGHT, and lines BELOW the cursor, then move text on a line-by-line basis to replace deleted material.

## ESCAPE SEQUENCES AND CONTROL CHARACTERS

### A.10 TAB-SETTING OPERATIONS

|      | <u>Function</u>                | <u>ZENITH</u> | <u>ANSI</u> | <u>Global</u>    |
|------|--------------------------------|---------------|-------------|------------------|
|      | Enable user tab stops          |               |             | ESC : 2 h        |
|      | Enable standard tab stops      |               |             | ESC : 2 l        |
| (18) | Set tab stop at current column |               | ESC H       | ESC : s          |
|      | Set tab stops at spec. columns |               |             | ESC : 0 s        |
|      | Set tab stops every Pn columns |               |             | ESC : Pn ; Pn..s |
|      | Clear all tab stops            |               | ESC [ 3 g   | ESC : c          |
|      | Clear tab at current column    |               | ESC [ g     | ESC : 0 c        |
|      | Clear tabs at spec. columns    |               | ESC [ 0 g   | ESC : Pc ; Pc..c |
|      | Move to previous stop          | ESC TAB       | ESC TAB     |                  |

Note 18 Tab stops may be set or cleared as desired with these sequences, but the settings will not become effective until 'user tabs' are enabled by the global sequence shown above.

### A.11 SCROLLING OPERATIONS

|      | <u>Function</u>                  | <u>ZENITH</u> | <u>ANSI</u>     | <u>Global</u>   |
|------|----------------------------------|---------------|-----------------|-----------------|
|      | Backward 1 page                  |               |                 | ESC : B         |
|      | Backward Pn lines                |               |                 | ESC : Pn B      |
|      | Forward 1 page                   |               |                 | ESC : F         |
|      | Forward Pn lines                 |               |                 | ESC : Pn F      |
|      | Move to start of OSM             |               |                 | ESC : 0 a       |
|      | Move to end of OSM               |               |                 | ESC : a         |
|      | Position screen home line in OSM |               |                 | ESC : Pn a      |
|      | Slow page scroll enabled         |               |                 | ESC : 3 h       |
|      | Slow page scroll disabled        |               |                 | ESC : 3 l       |
| (20) | Define scrolling region          |               | ESC [ Pt ; Pb r | ESC : Pt ; Pb r |
|      | Set relative origin mode         |               | ESC [ ? 6 h     | ESC : 16 h      |
|      | Set absolute origin mode         |               | ESC [ ? 6 l     | ESC : 16 l      |
|      | Set hold-screen mode             | ESC [         | ESC [ > 3 h     |                 |
|      |                                  | ESC x 3       |                 |                 |
|      | Clear hold-screen mode           | ESC \         | ESC [ > 3 l     |                 |
|      |                                  | ESC y 3       |                 |                 |

## ESCAPE SEQUENCES AND CONTROL CHARACTERS

Note 20 Interpretation of this sequence is affected by EDT-compatibility. See section 4.13.1.1. Text scrolled outside the defined region is lost; not put in OSM.

### A.12 SCREEN FORMATTING

|      | <u>Function</u>        | <u>ZENITH</u> | <u>ANSI</u> | <u>Global</u> |
|------|------------------------|---------------|-------------|---------------|
| (21) | Set 80-col format      |               | ESC [ ? 3 l | ESC : 1 f     |
|      | Set 132-col format     |               | ESC [ ? 3 h | ESC : 2 f     |
|      | Set 24-line format     |               |             | ESC : 3 f     |
|      | Set 49-line format     |               |             | ESC : 4 f     |
|      | Set non-reverse screen |               | ESC [ ? 5 l | ESC : 7 l     |
|      | Set reverse screen     |               | ESC [ ? 5 h | ESC : 7 h     |
|      | Enable status line     | ESC x 1       | ESC [ > 1 h |               |
|      | Disable status line    | ESC y 1       | ESC [ > 1 l |               |

Note 21 The global sequence may specify both line and column parameters in one sequence: (ESC : 1 ; 4 f) sets 49x80. The terminal will exit from Tektronix mode if necessary. The ANSI sequence keeps the current page length.

### A.13 GRAPHIC RENDITION SEQUENCES

|      | <u>Function</u>        | <u>ZENITH</u> | <u>ANSI</u>          | <u>Global</u> |
|------|------------------------|---------------|----------------------|---------------|
|      | Select normal video    | ESC q         | ESC [ m<br>ESC [ 0 m |               |
|      | Select reverse video   | ESC p         | ESC [ 7 m            |               |
|      | Select ZENITH char set |               |                      | ESC : 10 l    |
|      | Select VT100 char set  |               |                      | ESC : 10 h    |
| (22) | Select graphics chars  | ESC F         | ESC [ 10 m           |               |
|      | Select normal chars    | ESC G         | ESC [ 11 m           |               |
|      | Define GO as USASCII   | ESC ( B       | ESC ( B              |               |
|      | Define GO as graphics  | ESC ( O       | ESC ( O              |               |
|      | Define G1 as USASCII   | ESC ) B       | ESC ) B              |               |
|      | Define G1 as graphics  | ESC ) O       | ESC ) O              |               |



## ESCAPE SEQUENCES AND CONTROL CHARACTERS

Note 22 These sequences replace the lower-case characters with special graphics characters, either those used in the Z19 or those used in the VT100, as selected by the preceding global sequence. An alternative way to select 'character graphics' is to define G0 and G1 appropriately, and then use the SI control character to select the G1 definition and the S0 control character to select the G0 definition.

### A.14 PROGRAMMABLE FUNCTION KEYS

| <u>Function</u>              | <u>ZENITH</u> | <u>ANSI</u> | <u>Global</u>   |
|------------------------------|---------------|-------------|-----------------|
| Enable programmable keys     |               |             | ESC : 5 h       |
| Disable programmable keys    |               |             | ESC : 5 l       |
| Reset all function keys      |               |             | ESC : e         |
| Reset selected function keys |               |             | ESC : Pn ; Pn e |

In the 'reset' state the programmable keys send the sequences shown in section A.6. To program a given key, use:

ESC : Pn p delimiter string delimiter

where 'delimiter' is any character not occurring in the string. The function keys may be programmed when they are disabled but the default assignment will be used until they are activated. Pn selects the desired key according to the following table:

| Key   | Unshifted | Shifted |
|-------|-----------|---------|
| F1    | 1         | 9       |
| F2    | 2         | 10      |
| F3    | 3         | 11      |
| F4    | 4         | 12      |
| F5    | 5         | 13      |
| Blue  | 6         | 14      |
| Red   | 7         | 15      |
| White | 8         | 16      |

## ESCAPE SEQUENCES AND CONTROL CHARACTERS

### A.15 EXTERNAL ROM SEQUENCES

Form: ESC : Pn ; ...;Pn x      where

Pn(1) External ROM function desired (0-255.)

Pn(2) First optional parameter

...

Pn(15) Last optional parameter

A null parameter (two semicolons together) defaults to zero.  
Function codes 0-31 are reserved by Northwest Digital Systems.

### A.16 CONFIGURATION SEQUENCES

| <u>Function</u>                 | <u>ZENITH</u> | <u>ANSI</u>      | <u>Global</u> |
|---------------------------------|---------------|------------------|---------------|
| Reset to power-up configuration | ESC z         | ESC c<br>ESC [ z |               |
| Cursor off                      | ESC x 5       | ESC [ > 5 h      | ESC : 15 h    |
| Cursor on                       | ESC y 5       | ESC [ > 5 l      | ESC : 15 l    |
| Block cursor                    | ESC x 4       | ESC [ > 4 h      | ESC : 13 h    |
| Underscore cursor               | ESC y 4       | ESC [ > 4 l      | ESC : 13 l    |
| Non-blinking cursor             |               |                  | ESC : 8 h     |
| Blinking cursor                 |               |                  | ESC : 8 l     |
| Screen off                      |               |                  | ESC : 6 h     |
| Screen on                       |               |                  | ESC : 6 l     |
| Screen timeout disabled         |               |                  | ESC : 4 h     |
| Screen timeout enabled          |               |                  | ESC : 4 l     |
| Key Click enabled               | ESC y 2       | ESC [ > 2 l      |               |
| Key Click disabled              | ESC x 2       | ESC [ > 2 h      |               |
| Generate a Key Click            |               |                  | ESC : Pn T    |
| (24) Keyboard disabled          | ESC }         | ESC [ 2 h        |               |
| Keyboard enabled                | ESC [         | ESC [ 2 l        |               |
| Repeat key enabled              |               | ESC [ ? 8 h      |               |
| Repeat key disabled             |               | ESC [ ? 8 l      |               |
| Smart Shift key                 |               |                  | ESC : 9 h     |
| Dumb Shift key                  |               |                  | ESC : 9 l     |
| Auto LF on CR                   | ESC x 8       | ESC [ > 8 h      |               |
| No Auto LF on CR                | ESC y 8       | ESC [ > 8 l      |               |

## ESCAPE SEQUENCES AND CONTROL CHARACTERS

| <u>Function</u>  | <u>ZENITH</u> | <u>ANSI</u>               | <u>Global</u> |
|--|---------------|---------------------------|---------------|
| Auto CR on LF  | ESC x 9       | ESC [ > 9 h<br>ESC [ 20 h |               |
| No Auto CR on LF   | ESC y 9       | ESC [ > 9 L<br>ESC [ 20 L |               |
| Wrap at end-of-line  | ESC v         | ESC [ ? 7 h               |               |
| Discard at end-of-line   | ESC w         | ESC [ ? 7 L               |               |
| Smart Wrap   |               |                           | ESC : 12 h    |
| Dumb Wrap  |               |                           | ESC : 12 L    |
| Full duplex  |               |                           | ESC : 14 h    |
| Half Duplex  |               |                           | ESC : 14 L    |
| Xmit speed limiting enabled  |               |                           | ESC : 1 h     |
| Xmit speed limiting disabled   |               |                           | ESC : 1 L     |
| Output Sync enabled  |               |                           | ESC : 17 h    |
| Output Sync disabled   |               |                           | ESC : 17 L    |
| (20) Set Baud Rate   | ESC r Bn      | ESC [ Pn r                |               |
| Baud rate codes (Bn,Pn): 110=(A,1), 150=(B,2), 300=(C,3)<br>600=(D,4), 1200=(E,5), 1800=(F,6), 2000=(G,7), 2400=(H,8)<br>3600=(I,9), 4800=(J,10), 7200=(K,11), 9600=(L,12), 19200=(M,13) |               |                           |               |

Note 24 The keyboard is enabled by entering SETUP mode

### A.17 TEKTRONIX MODE ESCAPE SEQUENCES

|           |  |
|-----------|--|
| ESC FF    | Clear the screen and home the cursor.              |
| ESC : 5 f | Move cursor to home without clearing screen.       |
| ESC : E   | Erase the screen without moving the cursor.        |
| ESC ENQ   | Request status info (see 4.6.1.3, 4.6.2.4, 4.6.3). |
| ESC SUB   | Enter GIN state (see 4.6.3).                       |
| ESC DEL   | Change the data level to black.                    |
| ESC a     | Change the data level to white.                    |
| ESC ETB   | Activate the optional hardcopy device.             |

## ESCAPE SEQUENCES AND CONTROL CHARACTERS

ESC X Set GIN-state cursor to the current alpha or vector cursor position, but do not go into GIN state. This allows the program to 'pre-position' the cross-hair cursor.

Scroll Graphics Region ESC : Pb ; Pt ; Pd; S where

Pb = Bottom line in Screen Coordinates (0-249)

Pt = Top line in Screen Coordinates (0-249)

Pd = Direction: 0 = left, 1 = right

This sequence will shift the specified region by 8 screen coordinates either to the left or to the right. To change from 'Tekpoints' to 'screen coordinates' use:

X' = X/2

X = horizontal position in tekpoints

Y' = (Y\*41+18)/128 Y = vertical position in tekpoints

### A.18 MISCELLANEOUS ITEMS

|      | <u>FUNCTION</u>      | <u>ZENITH</u> | <u>ANSI</u> |
|------|----------------------|---------------|-------------|
| (25) | Transmit page        | ESC #         | ESC [ p     |
| (26) | Transmit status line | ESC ]         | ESC [ q     |
| (25) | Transmit cursor line | ESC CR        | ESC CR      |
| (27) | Generate hardcopy    | ESC ETB       | ESC ETB     |

Note 25 All information on the line (screen) is transmitted, including the escape sequences required to switch to reverse video, etc. The screen is treated as one continuous memory and no EOL codes are generated, except in the case of (ESC CR) which skips trailing blanks and ends each line with a CR.

Note 26 The status line must be enabled for this function to work.

Note 27 Requires auxiliary I/O board

### A.19 IGNORED VT100 ESCAPE SEQUENCES

Any ANSI escape sequence of the form: (ESC # Pn)

Any ANSI escape sequence of the form: (ESC [ 2 ; Pn y)

Any ANSI set/reset sequence other than those listed

Send device attributes: ESC Z

## ESCAPE SEQUENCES AND CONTROL CHARACTERS

### A.20 SUMMARY OF SEQUENCES WITH MULTIPLE PARAMETERS

ESC : Pn ; Pn ; ... c Clear selected user tab stops

ESC : Pn ; Pn ; ... e Reset selected function keys

ESC : Pn ; Pn ; ... f Change format according to table below and  
reset any defined scrolling region

|   |                            |
|---|----------------------------|
| 1 | 80 columns                 |
| 2 | 132 columns                |
| 3 | 24 lines                   |
| 4 | 49 lines                   |
| 5 | Tektronix mode Alpha state |
| 6 | Tektronix mode Graph state |
| 7 | Easy-Entry graphics mode   |
| 8 | Previous text mode         |

ESC : Pn ; Pn ; ... h Global 'set' functions

|    |                                |
|----|--------------------------------|
| 1  | Enable transmit-speed limiting |
| 2  | Enable user tab stops          |
| 3  | Enable slow page scroll        |
| 4  | Disable screen timeout         |
| 5  | Enable programmable keys       |
| 6  | Turn screen off                |
| 7  | Set reverse screen             |
| 8  | Set non-blinking cursor        |
| 9  | Enable smart-shift key         |
| 10 | Select VT100 character set     |
| 11 | Enable EDT-compatibility       |
| 12 | Enable smart-wrap              |
| 13 | Set block cursor               |
| 14 | Set full-duplex                |
| 15 | Disable cursor                 |
| 16 | Set relative-origin mode       |
| 17 | Enable output synchronization  |
| 18 | Set ANSI escape sequence mode  |

## ESCAPE SEQUENCES AND CONTROL CHARACTERS

ESC : Pn ; Pn ; ... L Global 'reset' functions

- 1 Disable transmit-speed limiting
- 2 Disable user tab stops
- 3 Disable slow page scroll
- 4 Enable screen timeout
- 5 Disable programmable keys
- 6 Turn screen on
- 7 Set non-reverse screen
- 8 Set blinking cursor
- 9 Disable smart-shift key
- 10 Select Zenith character set
- 11 Disable EDT-compatibility
- 12 Disable smart wrap
- 13 Set underscore cursor
- 14 Set half-duplex
- 15 Enable cursor
- 16 Set absolute-origin mode
- 17 Disable output synchronization
- 18 Set ZENITH escape sequence mode

ESC : Pn ; Pn ; ... s Set user tab stops

ESC : Pn ; Pn ; ... x Send external ROM control message

ESC [ Pn ; Pn ; ... h ANSI 'set' functions

- 2 Disable keyboard
- 4 Enable insert-character state
- 20 Enable auto CR on LF

ESC [ > Pn ; Pn ; ... h Zenith private ANSI 'set' functions

- 1 Enable status line (line 25 or 50)
- 2 Disable key click
- 3 Enable hold-screen mode
- 4 Enable block cursor
- 5 Disable cursor
- 6 Set shifted keypad
- 7 Set keypad alternate mode
- 8 Enable auto LF on CR
- 9 Enable auto CR on LF

## ESCAPE SEQUENCES AND CONTROL CHARACTERS

ESC [ ? Pn ; Pn ; ... h DEC private ANSI 'set' functions

- 1 Set ANSI cursor keys alternate mode
- 2 Set ZENITH mode (not VT100 compatible)
- 3 Set 132-column display
- 5 Set reverse screen
- 6 Set relative-origin mode
- 7 Enable auto-wrap at end-of-line
- 8 Enable repeat key

ESC [ Pn ; Pn ; ... L ANSI 'reset' functions

- 2 Enable keyboard
- 4 Disable insert-character state
- 20 Disable auto CR on LF

ESC [ > Pn ; Pn ; ... L Zenith private ANSI 'reset' functions

- 1 Disable status line (line 25 or 50)
- 2 Disable key click
- 3 Disable hold-screen mode
- 4 Set underscore cursor
- 5 Enable cursor
- 6 Set unshifted keypad
- 7 Disable keypad alternate mode
- 8 Disable auto LF on CR
- 9 Disable auto CR on LF

ESC [ ? Pn ; Pn ; ... L DEC private ANSI 'reset' functions

- 1 Set ANSI cursor keys normal mode
- 2 Set ZENITH (VT52) escape sequence mode
- 3 Set 80-column mode
- 5 Set non-reverse screen
- 6 Set absolute-origin mode
- 7 Discard characters at EOL
- 8 Disable repeat key

ESC [ Pn ; Pn ; ... m Set Graphic Rendition

- 0 Normal video
- 7 Reverse video
- 10 Character graphics selected
- 11 Normal characters selected

## ESCAPE SEQUENCES AND CONTROL CHARACTERS

### A.21 SUMMARY OF SEQUENCES RECOGNIZED IN ZENITH ESCAPE MODE

|                      |   |
|----------------------|---|
| ESC TAB              | Backwards tab                                   |
| ESC FF               | Enter Tektronix Alpha state and erase screen    |
| ESC CR               | Transmit cursor line (with attributes) plus CR  |
| ESC ETB              | Print page on hardcopy device                   |
| ESC #                | Transmit page                                   |
| ESC ( B              | Define G0 to be the USASCII character set       |
| ESC ( O              | Define G0 to be char graphics (Zenith or VT100) |
| ESC ) B              | Define G1 to be the USASCII character set       |
| ESC ) O              | Define G1 to be char graphics (Zenith or VT100) |
| ESC 7                | Save cursor position                            |
| ESC 8                | Restore cursor position                         |
| ESC : B              | Scroll backward 1 page into off-screen memory   |
| ESC : Pn B           | Scroll backward Pn lines into off-screen memory |
| ESC : E              | Erase all off-screen memory                     |
| ESC : O E            | Erase from cursor to end of off-screen memory   |
| ESC : F              | Scroll forward 1 page into off-screen memory    |
| ESC : Pn F           | Scroll forward Pn lines into off-screen memory  |
| ESC : Pr ; Pc H      | Position cursor at (row,col) specified          |
| ESC : Pc L           | Enable status line and position cursor at (col) |
| ESC : P or ESC : O P | Clear defined scrolling region and home cursor  |
| ESC : 1 P            | Clear the entire screen and set cursor position |
| ESC : Pn T           | Generate variable-length tick                   |
| ESC : Pn a           | Set absolute home line in off-screen memory     |
| ESC : O c            | Clear user tab stop at current cursor position  |
| ESC : c              | Clear all user tab stops                        |
| ESC : Pn ; ...; Pn c | Clear selected user tab stops                   |
| ESC : e              | Reset all programmable function keys            |
| ESC : Pn ; ...; Pn e | Reset selected programmable function keys       |
| ESC : 1 f            | Enter 80-col text mode                          |
| ESC : 2 f            | Enter 132-col text mode                         |
| ESC : 3 f            | Enter 24-line text mode                         |
| ESC : 4 f            | Enter 49-line text mode                         |
| ESC : 1 ; 3 f        | Enter 24x80 format, etc.                        |
| ESC : 5 f            | Enter Tektronix mode Alpha state                |
| ESC : 6 f            | Enter Tektronix mode Graph state                |
| ESC : 7 f            | Enter Easy-Entry graphics mode                  |
| ESC : 8 f            | Enter previous text mode from Tektronix mode    |
| ESC : 1 h            | Enable transmit-speed limiting                  |
| ESC : 2 h            | Enable user tab stops                           |
| ESC : 3 h            | Enable slow page scroll                         |
| ESC : 4 h            | Disable screen timeout                          |
| ESC : 5 h            | Enable programmable function keys               |
| ESC : 6 h            | Turn screen off                                 |
| ESC : 7 h            | Set reverse screen                              |
| ESC : 8 h            | Set non-blinking text-mode cursor               |
| ESC : 9 h            | Enable smart shift key                          |
| ESC : 10 h           | Select VT100 character set                      |
| ESC : 11 h           | Enable EDT-compatibility                        |



## ESCAPE SEQUENCES AND CONTROL CHARACTERS

|                      |  |
|----------------------|--|
| ESC : 12 h           | Enable smart-wrap  |
| ESC : 13 h           | Set block cursor   |
| ESC : 14 h           | Set full-duplex  |
| ESC : 15 h           | Disable cursor   |
| ESC : 16 h           | Set relative-origin mode                                     |
| ESC : 17 h           | Enable output synchronization (XON/XOFF)                     |
| ESC : 18 h           | Set ANSI escape sequence mode                                |
| ESC : 1 L            | Disable transmit speed limiting                              |
| ESC : 2 L            | Disable user tab stops                                       |
| ESC : 3 L            | Disable slow page scroll                                     |
| ESC : 4 L            | Enable screen timeout  |
| ESC : 5 L            | Disable programmable function keys                           |
| ESC : 6 L            | Turn screen on   |
| ESC : 7 L            | Set non-reverse screen                                       |
| ESC : 8 L            | Set blinking text mode cursor                                |
| ESC : 9 L            | Disable smart shift key                                      |
| ESC : 10 L           | Select Zenith character set                                  |
| ESC : 11 L           | Disable EDT-compatibility                                    |
| ESC : 12 L           | Disable smart-wrap   |
| ESC : 13 L           | Set underscore cursor  |
| ESC : 14 L           | Set half-duplex  |
| ESC : 15 L           | Enable cursor  |
| ESC : 16 L           | Set absolute-origin mode                                     |
| ESC : 17 L           | Disable output synchronization (XON/XOFF)                    |
| ESC : 18 L           | Set ZENITH escape sequence mode                              |
| ESC : 6 n            | Report cursor position. Response is:<br>ESC : Pr ; Pc R      |
| ESC : Pn p * text *  | Program function key (* is any unique char)                  |
| ESC : q              | Report screen format. Response is:<br>ESC : Pn q             |
| ESC : Pt ; Pb r      | Define scrolling region                                      |
| ESC : s              | Set user tab stop at current cursor position                 |
| ESC : Pn ; ...; Pn s | Set selected user tab stops                                  |
| ESC : Pn t           | Set multiple tab stops                                       |
| ESC : w              | Report text-mode window position. Response is:<br>ESC : Pn a |
| ESC : Pn ; ...; Pn x | Send external ROM control message                            |
| ESC <                | Set ANSI escape sequence mode                                |
| ESC =                | Enable keypad alternate mode                                 |
| ESC >                | Disable keypad alternate mode                                |
| ESC @                | Enable insert character state                                |
| ESC A                | Cursor up  |
| ESC B                | Cursor down  |
| ESC C                | Cursor right   |
| ESC D                | Cursor left  |
| ESC E                | Erase screen and home cursor                                 |
| ESC F                | Select graphics characters                                   |
| ESC G                | Select normal characters                                     |
| ESC H                | Move cursor to home position                                 |
| ESC I                | Reverse linefeed (index)                                     |

## ESCAPE SEQUENCES AND CONTROL CHARACTERS

|                       |  |
|-----------------------|--|
| ESC J                 | Erase from cursor to end of page   |
| ESC K                 | Erase from cursor to end of line   |
| ESC L                 | Insert line  |
| ESC M                 | Delete line  |
| ESC N                 | Delete character   |
| ESC O                 | Disable insert character state   |
| ESC Y rowchar colchar | Direct cursor addressing   |
| ESC Z                 | Request identification. Responds with<br>ESC / K (VT52)  |
| ESC [                 | Enable hold-screen mode  |
| ESC \                 | Disable hold-screen mode   |
| ESC ]                 | Transmit status line (line 25 or 50) IF enabled  |
| ESC b                 | Erase from cursor to beginning of display  |
| ESC j                 | Save cursor position   |
| ESC k                 | Restore cursor position  |
| ESC l                 | Erase entire line  |
| ESC n                 | Report cursor position. Response is<br>ESC Y rowchar colchar   |
| ESC o                 | Erase from cursor to beginning of line   |
| ESC p                 | Select reverse video   |
| ESC q                 | Select normal video  |
| ESC r Br              | Set baud rate. The final char (Br) is: A=110<br>B=150, C=300, D=600, E=1200, F=1800, G=2000<br>H=2400, I=3600, J=4800, K=7200, L=9600, M=19200 |
| ESC t                 | Set shifted keypad   |
| ESC u                 | Set unshifted keypad   |
| ESC v                 | Enable auto-wrap at end-of-line  |
| ESC w                 | Discard characters at end-of-line  |
| ESC x 1               | Enable status line (line 25 or 50)   |
| ESC x 2               | Disable key click  |
| ESC x 3               | Enable hold-screen mode  |
| ESC x 4               | Set block cursor   |
| ESC x 5               | Disable cursor   |
| ESC x 6               | Set shifted keypad   |
| ESC x 7               | Enable alternate keypad mode   |
| ESC x 8               | Enable auto LF on CR   |
| ESC x 9               | Enable auto CR on LF   |
| ESC y 1               | Disable status line (line 25 or 50)  |
| ESC y 2               | Enable key click   |
| ESC y 3               | Disable hold-screen mode   |
| ESC y 4               | Set underscore cursor  |
| ESC y 5               | Enable cursor  |
| ESC y 6               | Set unshifted keypad   |
| ESC y 7               | Disable alternate keypad mode  |
| ESC y 8               | Disable auto LF on CR  |
| ESC y 9               | Disable auto CR on LF  |
| ESC z                 | Reset to power-up state  |
| ESC {                 | Enable keyboard  |
| ESC }                 | Disable keyboard (enabled by going to SETUP)   |

## ESCAPE SEQUENCES AND CONTROL CHARACTERS

### A.22 SUMMARY OF SEQUENCES RECOGNIZED IN ANSI ESCAPE MODE

|                      |  |
|----------------------|--|
| ESC TAB              | Backwards tab                                    |
| ESC FF               | Enter Tektronix Alpha state and erase screen     |
| ESC CR               | Transmit cursor line (with attributes) plus CR   |
| ESC ETB              | Print page on hardcopy device                    |
| ESC ( B              | Define G0 to be the USASCII character set        |
| ESC ( O              | Define G0 to be char graphics (Zenith or VT100)  |
| ESC ) B              | Define G1 to be the USASCII character set        |
| ESC ) O              | Define G1 to be char graphics (Zenith or VT100)  |
| ESC 7                | Save cursor position                             |
| ESC 8                | Restore cursor position                          |
| ESC : B              | Scroll backward 1 page into off-screen memory    |
| ESC : Pn B           | Scroll backward Pn lines into off-screen memory  |
| ESC : E              | Erase all off-screen memory                      |
| ESC : O E            | Erase from cursor to end of off-screen memory    |
| ESC : F              | Scroll forward 1 page into off-screen memory     |
| ESC : Pn F           | Scroll forward Pn lines into off-screen memory   |
| ESC : Pr ; Pc H      | Position cursor at (row,col) specified           |
| ESC : Pc L           | Enable status line and position cursor at (col)  |
| ESC : P or ESC : O P | Clear defined scrolling region and home cursor   |
| ESC : 1 P            | Clear the entire screen and set cursor position  |
| ESC : Pn T           | Generate variable-length click                   |
| ESC : Pn a           | Set absolute home line in off-screen memory      |
| ESC : O c            | Clear user tab stop at current cursor position   |
| ESC : c              | Clear all user tab stops                         |
| ESC : Pn ; ...; Pn c | Clear selected user tab stops                    |
| ESC : e              | Reset all programmable function keys             |
| ESC : Pn ; ...; Pn e | Reset selected programmable function keys        |
| ESC : 1 f            | Enter 80-col text mode                           |
| ESC : 2 f            | Enter 132-col text mode                          |
| ESC : 3 f            | Enter 24-line text mode                          |
| ESC : 4 f            | Enter 49-line text mode                          |
| ESC : 1 ; 3 f        | Enter 24x80 format, etc.                         |
| ESC : 5 f            | Enter Tektronix mode Alpha state and home cursor |
| ESC : 6 f            | Enter Tektronix mode Graph state                 |
| ESC : 7 f            | Enter Easy-Entry graphics mode                   |
| ESC : 8 f            | Enter previous text mode from Tektronix mode     |
| ESC : 1 h            | Enable transmit-speed limiting                   |
| ESC : 2 h            | Enable user tab stops                            |
| ESC : 3 h            | Enable slow page scroll                          |
| ESC : 4 h            | Disable screen timeout                           |
| ESC : 5 h            | Enable programmable function keys                |
| ESC : 6 h            | Turn screen off                                  |
| ESC : 7 h            | Set reverse screen                               |
| ESC : 8 h            | Set non-blinking text-mode cursor                |
| ESC : 9 h            | Enable smart shift key                           |
| ESC : 10 h           | Select VT100 character set                       |
| ESC : 11 h           | Enable EDT-compatibility                         |
| ESC : 12 h           | Enable smart-wrap                                |

## ESCAPE SEQUENCES AND CONTROL CHARACTERS

|                      |   |
|----------------------|---|
| ESC : 13 h           | Set block cursor  |
| ESC : 14 h           | Set full-duplex   |
| ESC : 15 h           | Disable cursor  |
| ESC : 16 h           | Set relative origin mode                                |
| ESC : 17 h           | Enable output synchronization (XON/XOFF)                |
| ESC : 18 h           | Set ANSI escape sequence mode                           |
| ESC : 1 L            | Disable transmit speed limiting                         |
| ESC : 2 L            | Disable user tab stops                                  |
| ESC : 3 L            | Disable slow page scroll                                |
| ESC : 4 L            | Enable screen timeout                                   |
| ESC : 5 L            | Disable programmable function keys                      |
| ESC : 6 L            | Turn screen on  |
| ESC : 7 L            | Set non-reverse screen                                  |
| ESC : 8 L            | Set blinking text mode cursor                           |
| ESC : 9 L            | Disable smart shift key                                 |
| ESC : 10 L           | Select Zenith character set                             |
| ESC : 11 L           | Disable EDT-compability                                 |
| ESC : 12 L           | Disable smart-wrap                                      |
| ESC : 13 L           | Set underscore cursor                                   |
| ESC : 14 L           | Set half-duplex   |
| ESC : 15 L           | Enable cursor   |
| ESC : 16 L           | Set absolute origin mode                                |
| ESC : 17 L           | Disable output synchronization (XON/XOFF)               |
| ESC : 18 L           | Set ZENITH escape sequence mode                         |
| ESC : 6 n            | Report cursor position. Response is:<br>ESC : Pr ; Pc R |
| ESC : Pn p * text *  | Program function key (* is any unique char)             |
| ESC : q              | Report screen format. Response is:<br>ESC : Pn q        |
| ESC : Pt ; Pb r      | Define scrolling region                                 |
| ESC : s              | Set user tab stop at current cursor position            |
| ESC : Pn ; ...; Pn s | Set selected user tab stops                             |
| ESC : Pn t           | Set multiple tab stops                                  |
| ESC : w              | Report text-mode window. Response is:<br>ESC : Pn a     |
| ESC : Pn ; ...; Pn x | Send external ROM control message                       |
| ESC =                | Enable keypad alternate mode                            |
| ESC >                | Disable keypad alternate mode                           |
| ESC D                | Move cursor down one line                               |
| ESC E                | Move cursor to start of next line                       |
| ESC H                | Set tab at current column                               |
| ESC M                | Move cursor up one line                                 |
| ESC c                | Reset to power-up state                                 |
| ESC [ Pn A           | Move cursor up Pn (1) lines                             |
| ESC [ Pn B           | Move cursor down Pn (1) lines                           |
| ESC [ Pn C           | Move cursor right Pn (1) lines                          |
| ESC [ Pn D           | Move cursor left Pn (1) lines                           |
| ESC [ Pr ; Pc H      | Position cursor to (row,col) (1,1)                      |
| ESC [ J or ESC [ O J | Erase from cursor to end of page                        |
| ESC [ 1 J            | Erase from beginning of page to cursor                  |

## ESCAPE SEQUENCES AND CONTROL CHARACTERS

|                      |   |
|----------------------|---|
| ESC [ 2 J            | Erase entire screen (if EDT, don't move cursor)                         |
| ESC [ K or ESC [ O K | Erase from cursor to end of line  |
| ESC [ 1 K            | Erase from beginning of line to cursor                                  |
| ESC [ 2 K            | Erase entire line   |
| ESC [ Pn L           | Insert Pn (1) lines   |
| ESC [ Pn M           | Delete Pn (1) lines from cursor forward                                 |
| ESC [ Pn P           | Delete Pn (1) characters from cursor forward                            |
| ESC [ c or ESC [ O c | Report device attributes. Responds with<br>ESC [ ? 1 ; 0 c (no options) |
| ESC [ Pr ; Pc f      | Same as ESC [ Pr ; Pc H (position cursor)                               |
| ESC [ g or ESC [ O g | Clear tab at current column position                                    |
| ESC [ 3 g            | Clear all user tab stops  |
| ESC [ 2 h            | Disable keyboard (re-enabled by Set-up)                                 |
| ESC [ 4 h            | Enable insert-character state   |
| ESC [ 20 h           | Enable auto CR on LF  |
| ESC [ > 1 h          | Enable status line (line 25 or 50)                                      |
| ESC [ > 2 h          | Disable key click   |
| ESC [ > 3 h          | Enable hold-screen mode   |
| ESC [ > 4 h          | Set block cursor  |
| ESC [ > 5 h          | Disable cursor  |
| ESC [ > 6 h          | Set shifted keypad  |
| ESC [ > 7 h          | Set keypad alternate mode (same as ESC =)                               |
| ESC [ > 8 h          | Enable auto LF on CR  |
| ESC [ > 9 h          | Enable auto CR on LF  |
| ESC [ ? 1 h          | Set ANSI cursor keys alternate mode                                     |
| ESC [ ? 2 h          | Set ZENITH esc seq mode (not VT100 compatible)                          |
| ESC [ ? 3 h          | Set 132-column display  |
| ESC [ ? 5 h          | Set reverse screen  |
| ESC [ ? 6 h          | Set relative origin mode  |
| ESC [ ? 7 h          | Enable auto-wrap at end-of-line   |
| ESC [ ? 8 h          | Enable repeat key   |
| ESC [ 2 l            | Enable keyboard   |
| ESC [ 4 l            | Disable insert-character state  |
| ESC [ 20 l           | Disable auto CR on LF   |
| ESC [ > 1 l          | Disable status line (line 25 or 50)                                     |
| ESC [ > 2 l          | Enable key click  |
| ESC [ > 3 l          | Disable hold-screen mode  |
| ESC [ > 4 l          | Enable underscore cursor  |
| ESC [ > 5 l          | Enable cursor   |
| ESC [ > 6 l          | Set unshifted keypad  |
| ESC [ > 7 l          | Exit keypad alternate mode (same as ESC >)                              |
| ESC [ > 8 l          | Disable auto LF on CR   |
| ESC [ > 9 l          | Disable auto CR on LF   |
| ESC [ ? 1 l          | Set ANSI cursor keys normal mode  |
| ESC [ ? 2 l          | Set ZENITH escape sequence mode   |
| ESC [ ? 3 l          | Set 80-column screen format   |
| ESC [ ? 5 l          | Set non-reverse screen  |
| ESC [ ? 6 l          | Set absolute origin mode  |
| ESC [ ? 7 l          | Disable auto-wrap at end-of-line  |
| ESC [ ? 8 l          | Disable repeat key  |

## ESCAPE SEQUENCES AND CONTROL CHARACTERS

|                                 |   |
|---------------------------------|---|
| ESC [ m or ESC [ O m            | Disable reverse video   |
| ESC [ 7 m                       | Set reverse video attribute   |
| ESC [ 10 m                      | Select graphics characters  |
| ESC [ 11 m                      | Select normal characters  |
| ESC [ 5 n                       | Report device status. Response is<br>ESC [ O n (Ready, terminal OK)   |
| ESC [ 6 n                       | Report cursor position. Response is<br>ESC [ Pr ; Pc R (row,col) location   |
| ESC [ p                         | Transmit page   |
| ESC [ q                         | Transmit status line (line 25 or 50) IF enabled   |
| ESC [ Pn r<br>(not in EDT mode) | Change baud rate. Permissible Pn values:<br>1=110, 2=150, 3=300, 4=600, 5=1200,<br>6=1800, 7=2000 8=2400, 9=3600, 10=4800,<br>11=7200, 12=9600, 13=19200          |
| ESC [ Pt ; Pb r                 | Set scrolling region (Pt is top line, Pb is<br>bottom - both parameters are necessary in<br>non-EDT mode to distinguish this sequence from<br>'change baud rate') |
| ESC [ s                         | Save cursor position  |
| ESC [ u                         | Restore cursor position   |
| ESC [ z                         | Reset to power-up state (same as ESC c)   |

## ESCAPE SEQUENCES AND CONTROL CHARACTERS

### A.23 SUMMARY OF SEQUENCES RECOGNIZED IN TEKTRONIX AND EASY-ENTRY MODES

|                      |  |
|----------------------|--|
| ESC ENQ              | Request terminal-status report                   |
| ESC FF               | Clear screen and home the cursor                 |
| ESC ETB              | Print screen on hardcopy device                  |
| ESC SUB              | Enable cross-hair cursor and enter GIN state     |
| ESC DEL              | Change data level to black                       |
| ESC X                | Set GIN cursor pos. to current vector endpoint   |
| ESC a                | Change data level to white                       |
| ESC : E              | Erase screen without changing cursor position    |
| ESC : P              | Clear screen and home cursor (same as ESC FF)    |
| ESC : Pn P           | Same as ESC : P (and ERASE key function)         |
| ESC : Pb ; Pt ; Pd S | Scroll graphics region                           |
| ESC : Pn T           | Generate variable-length tick                    |
| ESC : O c            | Clear user tab stop at current cursor position   |
| ESC : c              | Clear all user tab stops                         |
| ESC : Pn ; ...; Pn c | Clear selected user tab stops                    |
| ESC : e              | Reset all programmable function keys             |
| ESC : Pn ; ...; Pn e | Reset selected programmable function keys        |
| ESC : 1 f            | Exit graphics mode, go to 80-col text mode       |
| ESC : 2 f            | Exit graphics mode, go to 132-col text mode      |
| ESC : 3 f            | Exit graphics mode, go to 24-line text mode      |
| ESC : 4 f            | Exit graphics mode, go to 49-line text mode      |
| ESC : 1 ; 3 f        | Exit graphics mode, go to 24x80 format, etc.     |
| ESC : 5 f            | Enter Tektronix mode Alpha state and home cursor |
| ESC : 6 f            | Enter Tektronix mode Graph state                 |
| ESC : 7 f            | Enter Easy-Entry graphics mode                   |
| ESC : 8 f            | Enter previous text mode from Tektronix mode     |
| ESC : 1 h            | Enable transmit-speed limiting                   |
| ESC : 2 h            | Enable user tab stops                            |
| ESC : 3 h            | Enable slow page scroll                          |
| ESC : 4 h            | Disable screen timeout                           |
| ESC : 5 h            | Enable programmable function keys                |
| ESC : 6 h            | Turn screen off                                  |
| ESC : 7 h            | Set reverse screen                               |
| ESC : 8 h            | Set non-blinking text-mode cursor                |
| ESC : 9 h            | Enable smart shift key                           |
| ESC : 10 h           | Select VT100 character set                       |
| ESC : 11 h           | Enable EDT-compatibility                         |
| ESC : 12 h           | Enable smart-wrap in text mode                   |
| ESC : 13 h           | Set block cursor in text mode                    |
| ESC : 14 h           | Set full-duplex                                  |
| ESC : 15 h           | Disable cursor (text-mode or Tektronix Alpha)    |
| ESC : 16 h           | Set relative origin mode                         |
| ESC : 17 h           | Enable output synchronization (XON/XOFF)         |
| ESC : 18 h           | Set ANSI escape sequence mode                    |
| ESC : 1 L            | Disable transmit speed limiting                  |
| ESC : 2 L            | Disable user tab stops                           |
| ESC : 3 L            | Disable slow page scroll                         |
| ESC : 4 L            | Enable screen timeout                            |

## ESCAPE SEQUENCES AND CONTROL CHARACTERS

|                      |  |
|----------------------|--|
| ESC : 5 L            | Disable programmable function keys               |
| ESC : 6 L            | Turn screen on                                   |
| ESC : 7 L            | Set non-reverse screen                           |
| ESC : 8 L            | Set blinking text mode cursor                    |
| ESC : 9 L            | Disable smart shift key                          |
| ESC : 10 L           | Select Zenith character set                      |
| ESC : 11 L           | Disable EDT-compability                          |
| ESC : 12 L           | Disable smart-wrap in text mode                  |
| ESC : 13 L           | Set underscore cursor in text mode               |
| ESC : 14 L           | Set half-duplex                                  |
| ESC : 15 L           | Enable cursor (text-mode or Tektronix Alpha)     |
| ESC : 16 L           | Set absolute origin mode                         |
| ESC : 17 L           | Disable output synchronization (XON/XOFF)        |
| ESC : 18 L           | Set ZENITH escape sequence mode                  |
| ESC : 6 n            | Report cursor position (meaningless response)    |
| ESC : Pn p * text *  | Program function key (* is any unique char)      |
| ESC : q              | Report screen format. Response is:<br>ESC : Pn q |
| ESC : s              | Set use tab stop at current cursor position      |
| ESC : Pn ; ...; Pn s | Set selected text mode user tab stops            |
| ESC : Pn t           | Set multiple tab stops                           |
| ESC : w              | Report text mode window (meaningless response)   |
| ESC : Pn ; ...; Pn x | Send external ROM control message                |



APPENDIX B  
VECTOR DRAWING TIMES

All coordinates are specified in terms of dot positions rather than tekpoints.

$X' = X/2$                       Tektronix X coordinates to dot position  
 $Y' = (Y*41+18)/128$       Tektronix Y coordinates to dot position

The symbols dx and dy represent the absolute value of the distance between two points on the X or Y axis of the screen.

$dx = \text{ABS} (X1' - X2')$   
 $dy = \text{ABS} (Y1' - Y2')$

The following times include the time to store and retrieve the characters from the input buffer (all times are in microseconds unless otherwise specified):

Vector Specification Time - t(vs)

|          |      |           |     |          |
|----------|------|-----------|-----|----------|
| For each | Hi Y | specified | add | 575 usec |
|          | Lo Y | "         | "   | 548 usec |
|          | Hi X | "         | "   | 580 usec |
|          | Lo X | "         | "   | 726 usec |

Dark Vectors:

For each dark vector add 650 usec to t(vs) (for GS processing).  
 Dark vectors do not require any drawing time.

## VECTOR DRAWING TIMES

### Light vectors:

Vertical Vectors:  $t(vs) + 355 + 29.8*(dy+1)$

Horizontal Vectors:  $t(vs) + 369 + (dx+1)*3.25 + \text{MIN}(dx+1,14)*47$

X-major Vectors ( $dx > dy$ ):

$t(vs) + 539 + (dx+1)*40.1 + \text{INT}((dx+1)/8)*7.6 + (dy+1)*16.25$

If vector is black add 10.8 usec.

Y-major Vectors: ( $dy > dx$ ):

$t(vs) + 528 + (dy+1)*38.5 + (dx+1)*13.5 + \text{INT}((dx+1)/8)*7.6$

If vector is black add 10.8 usec.

### Erase Vectors:

Add 650 usec to above vector times for erase vectors (for RS processing).

### Changing Data Level:

Add 1084 usec for changing data level (Esc a or Esc Rubout)

### Area Fill and Erase:

$t(vs) + 1084 + \{(dy+1) * [106 + (dx+1)*3.25 + \text{MIN}(dx+1,14)*47]\}$

### Screen Erase:

64 msec plus up to 33 msec for screen on/off timing

## APPENDIX C

### COMPOSITE VIDEO GENERATOR

The GP-19 logic board has provisions for connecting a composite video generator to drive an external video monitor. Socket U3 on the GP-19 provides the same signal pinouts as does socket U479 on the Z19 logic board (socket U406 on Z19-CNs). Composite video generators designed to operate with Z-19 terminals by connecting to socket U479 (or U406 on a Z19-CN) will operate correctly when connected to socket U3 on the GP-19 logic board.

The model 77319 composite video generator available from Magnolia Microsystems Incorporated of Seattle Washington operates both with Z19 and Z19-CN terminals and with GRAPHICS-PLUS. For information on this product contact:

Magnolia Microsystems, Inc.  
2264 - 15th Avenue West  
Seattle, Washington  
(206) 285-7266  
(800) 426-2841

## APPENDIX D

### AUXILIARY I/O CONNECTOR

The Auxiliary I/O connector is designed to allow installation of an auxiliary logic card containing I/O interfaces as well as additional ROM and/or RAM. As can be seen in Figure D-1, the connector provides access to all the Z80 CPU data and address signals as well as the control signals asserted by the Z80 to perform memory and I/O read/write operations. In addition, the following signals are also accessible:

1. EXT DEV L (from GP-19) A decoded I/O select signal which is asserted during Z80 I/O operations in the address range 78-7F.
2. EXT MEM L (from GP-19) A decoded memory select line that is asserted during Z80 memory read operations in the address range 3000-3FFF.
3. Z19 RST L (from GP-19) The reset signal from the Z-19 Logic board. This signal is asserted immediately after power-up and whenever the key combination RIGHT SHIFT-RESET is depressed.
4. EXT INT L (to GP-19) When low this signal asserts the INT\* input of the Z80 CPU.
5. LP STROBE (to G-19) This signal is connected to pin 3 of U50, the 6845 CRT controller.
6. 12.292 MHZ (from GP-19) This is a crystal controlled clock signal.
7. +5V (from GP-19) This is a regulated power supply. The load on this supply should be limited to 150 ma.
8. +16V (from GP-19) If the terminal is a Z19 then this is an unregulated power supply. If the terminal is a Z19-CN then this is a regulated 15V supply connected thru a 33 ohm series resistor. In either case the load on this supply should be limited to 25 ma.

## AUXILIARY I/O CONNECTOR

9. -12/-16V (from GP-19) If the terminal is a Z19 then this is an unregulated -16V power supply. If the terminal is a Z19-CN then this is a regulated -12V power supply. In either case the load should not exceed 25 ma.

FIGURE D-1

### Auxiliary I/O Connector Pinout

40 39

|           |     |   |   |            |   |
|-----------|-----|---|---|------------|---|
|           | GND | o | o | +5V        |   |
| CPU A14   | H   | o | o | CPU A13    | H |
| CPU D0    | H   | o | o | CPU A0     | H |
| CPU D1    | H   | o | o | CPU A1     | H |
| CPU D2    | H   | o | o | CPU A2     | H |
| CPU D3    | H   | o | o | CPU A3     | H |
| CPU D4    | H   | o | o | CPU A4     | H |
| CPU D5    | H   | o | o | CPU A5     | H |
| CPU D6    | H   | o | o | CPU A6     | H |
| CPU D7    | H   | o | o | CPU A7     | H |
| CPU RD    | L   | o | o | CPU A8     | H |
| CPU WR    | L   | o | o | CPU A9     | H |
| EXT DEV   | L   | o | o | CPU A10    | H |
| RESET     | L   | o | o | CPU A11    | H |
| EXT INT   | L   | o | o | CPU A12    | H |
| LP STB    | L   | o | o | EXT MEM    | L |
| CPU MREQ  | L   | o | o | 12.292 MHz |   |
| -12V/-16V |     | o | o | +15/+16V   |   |
| CPU A15   | H   | o | o | NC         |   |
| GND       |     | o | o | GND        |   |
|           |     | 2 | 1 |            |   |

APPENDIX E

VT100 GRAPHICS CHARACTER SET

When the VT100 character set is selected the terminal graphics character set is a replica of that used by VT100 compatible terminals. When graphics characters are enabled the special symbols shown in this appendix are displayed instead of the corresponding ASCII characters. In the following chart each graphics character is shown immediately below the corresponding ASCII character.

| _ | .     | a     | b     | c       | d   | e     | f   |
|---|-------|-------|-------|---------|-----|-------|-----|
|   | *     | * * * | * *   | *****   | *** | *     | *** |
|   | ***   | * *   | * *   | *       | *   | *     | * * |
|   | ***** | * * * | ***** | ***     | *   | *     | * * |
|   | ***** | * *   | * *   | *       | *** | ***** | *** |
|   | ***** | * * * | * *   | * ***** | *** | ***** |     |
|   | ***   | * *   | ***** | *       | * * | *     |     |
|   | *     | * * * | *     | ***     | *** | ***   |     |
|   |       |       | *     | *       | * * | *     |     |
|   |       |       | *     | *       | * * | *     |     |

VT100 GRAPHICS CHARACTER SET

| g     | h     | i     | j    | k    | l     | m     | n     |
|-------|-------|-------|------|------|-------|-------|-------|
|       | * *   | * *   | *    |      |       | *     | *     |
| *     | ** *  | * *   | *    |      |       | *     | *     |
| *     | * * * | * *   | *    |      |       | *     | *     |
| ***** | * **  | *     | **** | **** | ***** | ***** | ***** |
| *     | * *   | ***** |      | *    | *     |       | *     |
| *     | *     | *     |      | *    | *     |       | *     |
| ***** | *     | *     |      | *    | *     |       | *     |
|       | *     | *     |      | *    | *     |       | *     |
|       | ****  | *     |      | *    | *     |       | *     |

| o     | p     | q     | r     | s     | t     | u    | v     |
|-------|-------|-------|-------|-------|-------|------|-------|
| ***** |       |       |       |       | *     | *    | *     |
|       | ***** |       |       |       | *     | *    | *     |
|       |       | ***** |       |       | *     | *    | *     |
|       |       |       | ***** |       | ***** | **** | ***** |
|       |       |       |       |       | *     | *    |       |
|       |       |       |       |       | *     | *    |       |
|       |       |       |       | ***** | *     | *    |       |
|       |       |       |       |       | *     | *    |       |

| w     | x | y     | z     | {     |       | }     | ~      |
|-------|---|-------|-------|-------|-------|-------|--------|
|       | * |       |       |       |       | *     | **     |
|       | * |       | *     |       |       | *     | * *    |
|       | * | *     | *     |       |       | *     | *      |
| ***** | * | *     | *     | ***** | ***** | *     | *      |
| *     | * | *     | *     | *     | *     | ***** | *      |
| *     | * | *     | *     | *     | *     | ***** | *      |
| *     | * | *     | *     | *     | *     | *     | ****   |
| *     | * | *     | *     | *     | *     | *     | *** ** |
| *     | * | ***** | ***** |       |       | *     | *      |

APPENDIX F  
ASCII CHARACTERS

Hexadecimal Values of ASCII Characters

|   |   | Most Significant Digit |     |    |   |   |   |   |     |
|---|---|------------------------|-----|----|---|---|---|---|-----|
|   |   | 0                      | 1   | 2  | 3 | 4 | 5 | 6 | 7   |
| L | 0 | NUL                    | DLE | SP | 0 | @ | P | , | p   |
|   | 1 | SOH                    | DC1 | !  | 1 | A | Q | a | q   |
| A | 2 | STX                    | DC2 | "  | 2 | B | R | b | r   |
|   | 3 | ETX                    | DC3 | #  | 3 | C | S | c | s   |
| T | 4 | EOT                    | DC4 | \$ | 4 | D | T | d | t   |
|   | 5 | ENQ                    | NAK | %  | 5 | E | U | e | u   |
| S | 6 | ACK                    | SYN | &  | 6 | F | V | f | v   |
|   | 7 | BEL                    | ETB | '  | 7 | G | W | g | w   |
| G | 8 | BS                     | CAN | (  | 8 | H | X | h | x   |
|   | 9 | HT                     | EM  | )  | 9 | I | Y | i | y   |
| D | A | LF                     | SUB | *  | : | J | Z | j | z   |
|   | B | VT                     | ESC | +  | ; | K | [ | k | {   |
| I | C | FF                     | FS  | /  | < | L | \ | l |     |
|   | D | CR                     | GS  | -  | = | M | ] | m | }   |
| T | E | SO                     | RS  | .  | > | N | ^ | n | ~   |
|   | F | SI                     | US  | /  | ? | O | _ | o | DEL |



## ASCII CHARACTERS

| 7-BIT<br>OCTAL<br>CODE | DECIMAL<br>CODE | HEX<br>CODE | CHARACTER | CONTROL<br>KEYS | DESCRIPTION             |
|------------------------|-----------------|-------------|-----------|-----------------|-------------------------|
| 000                    | 0               | 00          | NUL       | @               | Null                    |
| 001                    | 1               | 01          | SOH       | A               | Start of Header         |
| 002                    | 2               | 02          | STX       | B               | Start of Text           |
| 003                    | 3               | 03          | ETX       | C               | End of Text             |
| 004                    | 4               | 04          | EOT       | D               | End of Transmission     |
| 005                    | 5               | 05          | ENQ       | E               | Enquiry (WRU)           |
| 006                    | 6               | 06          | ACK       | F               | Acknowledge (RU)        |
| 007                    | 7               | 07          | BEL       | G               | Bell                    |
| 010                    | 8               | 08          | BS        | H               | Backspace               |
| 011                    | 9               | 09          | HT        | I               | Horizontal Tab          |
| 012                    | 10              | 0A          | LF        | J               | Line Feed               |
| 013                    | 11              | 0B          | VT        | K               | Vertical Tab            |
| 014                    | 12              | 0C          | FF        | L               | Form Feed               |
| 015                    | 13              | 0D          | CR        | M               | Carriage Return         |
| 016                    | 14              | 0E          | SO        | N               | Shift Out               |
| 017                    | 15              | 0F          | SI        | O               | Shift In                |
| 020                    | 16              | 10          | DLE       | P               | Data Line Escape        |
| 021                    | 17              | 11          | DC1       | Q               | Device Control 1 (XON)  |
| 022                    | 18              | 12          | DC2       | R               | Device Control 2        |
| 023                    | 19              | 13          | DC3       | S               | Device Control 3 (XOFF) |
| 024                    | 20              | 14          | DC4       | T               | Device Control 4        |
| 025                    | 21              | 15          | NAK       | U               | Negative Acknowledge    |
| 026                    | 22              | 16          | SYN       | V               | Synchronous Idle (SYNC) |
| 027                    | 23              | 17          | ETB       | W               | End of Trans. Block     |
| 030                    | 24              | 18          | CAN       | X               | Cancel (CANCL)          |
| 031                    | 25              | 19          | EM        | Y               | End of Medium           |
| 032                    | 26              | 1A          | SUB       | Z               | Substitute              |
| 033                    | 27              | 1B          | ESC       | [               | Escape                  |
| 034                    | 28              | 1C          | FS        | \               | File Separator          |
| 035                    | 29              | 1D          | GS        | ]               | Group Separator         |
| 036                    | 30              | 1E          | RS        | ^               | Record Separator        |
| 037                    | 31              | 1F          | US        | _               | Unit Separator          |

## APPENDIX G

### OPTIONAL SOFTWARE

Northwest Digital Systems is assembling a library of software of interest to GRAPHICS-PLUS users. This software is available to interested users on an unsupported basis at a price intended only to defer the cost of distribution. The following items are presently included in the library:

1. PLOT-10 extensions: These FORTRAN routines are intended for users who already have copies of the Tektronix PLOT-10 Terminal Control System and Advanced Graphing II Library. These routines extend this library to take advantage of the data level, black vector, area erase, and area fill capabilities of GRAPHICS-PLUS.
2. Easy-Entry GIN support routines: These are routines written in both FORTRAN and BASIC intended for users who are employing the terminal in Easy-Entry graphics mode without a supporting library. These routines allow convenient use of Tektronix mode GIN state in conjunction with Easy-Entry graphics mode.
3. VAX/VMS EDT help library: This is a new help library for use with the EDT editor on VAX/VMS systems. This help library shows the correct keypad layout for Z19 terminals equipped with GRAPHICS-PLUS in both the VT52 (ZENITH) and VT100 (ANSI/EDT compatible) modes of operation. Also included is a command procedure for selectively invoking EDT with this help library assigned.

Please contact Northwest Digital Systems for a list of the software currently in the library, the available distribution media, and the cost of distribution.

## OPTIONAL SOFTWARE

The following program example represents the use of Easy-Entry Graphics Mode and GIN State. It is written to be compatible with MBASIC.

```

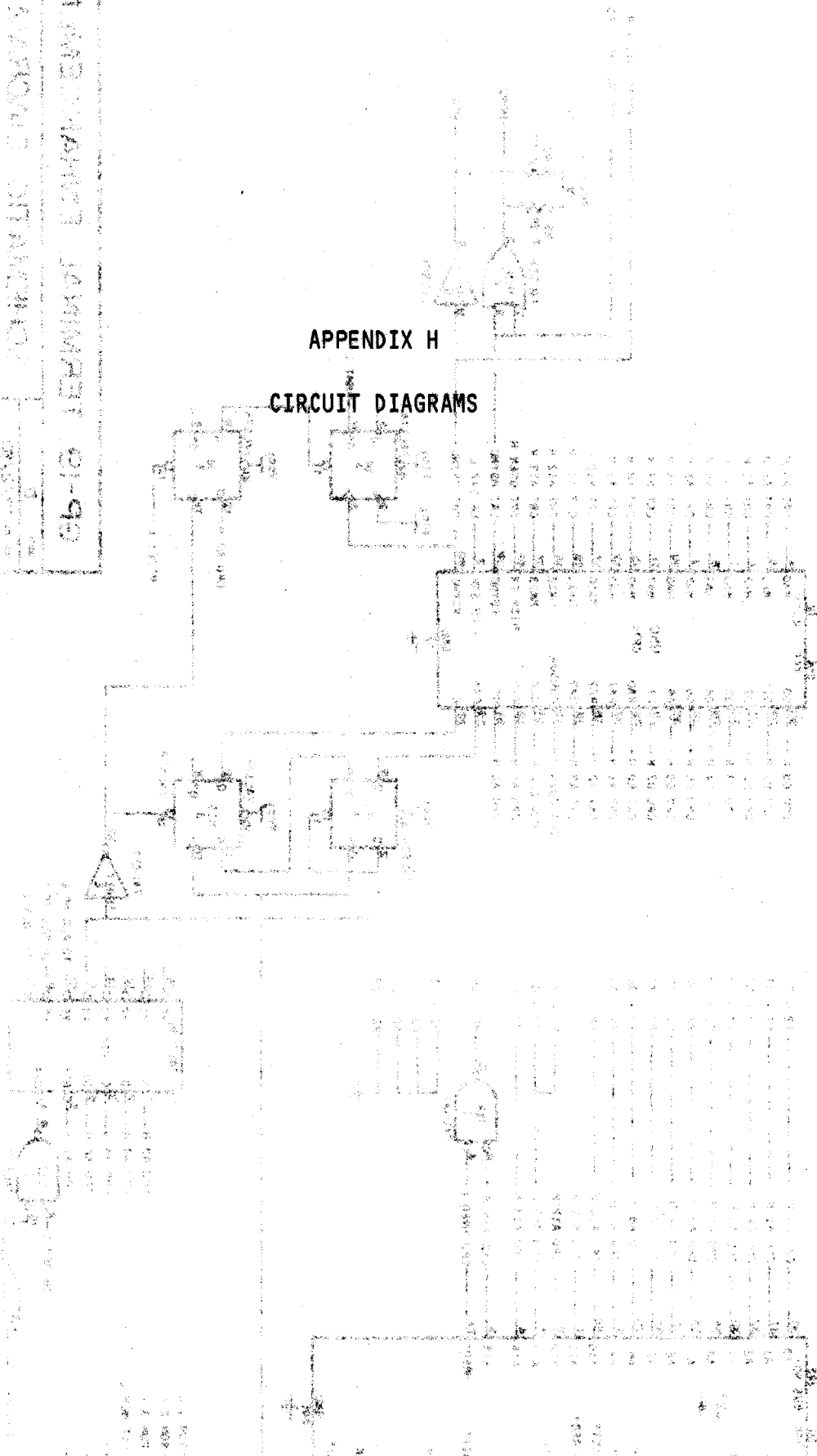
100  REM  GIN STATE ETCH-A-SKETCH
200  REM  Prints the pressed key and the cursor coordinates
300  REM  in "Easy-Entry" coordinates.
400  GOSUB 1000
500  PRINT "+0+0da ";TERM$;" ";X;" ";Y;
600  GOTO 400
700  REM
1000 REM  BASIC Subroutine to place the GP-19 in GIN state,
1010 REM  and then wait for the user to strike a key.
1020 REM  The X coordinate is returned as variable X.
1030 REM  The Y coordinate is returned as variable Y.
1040 REM  The terminating character is returned as variable TERM$.
1050 REM  This routine uses variables: ESC$,GIN$,GS$,SUB$,TERM$,X,Y
1060 REM  The GIN Trailer code must be set to send a CR.
1070 REM
1100 GS$=CHR$(29)
1110 ESC$=CHR$(27)
1120 SUB$=CHR$(26)
1130 PRINT ESC$;" :6f";ESC$;SUB$;
1140 LINE INPUT GIN$
1150 TERM$=MID$(GIN$,1,1)
1160 X=(ASC(MID$(GIN$,2,1))-32)*32 + (ASC(MID$(GIN$,3,1))-32)
1170 Y=(ASC(MID$(GIN$,4,1))-32)*32 + (ASC(MID$(GIN$,5,1))-32)
1180 X=INT(X/2)
1190 Y=INT((Y*41+18)/128)
1200 PRINT ESC$;" :7f";X;Y;"p"
1210 RETURN

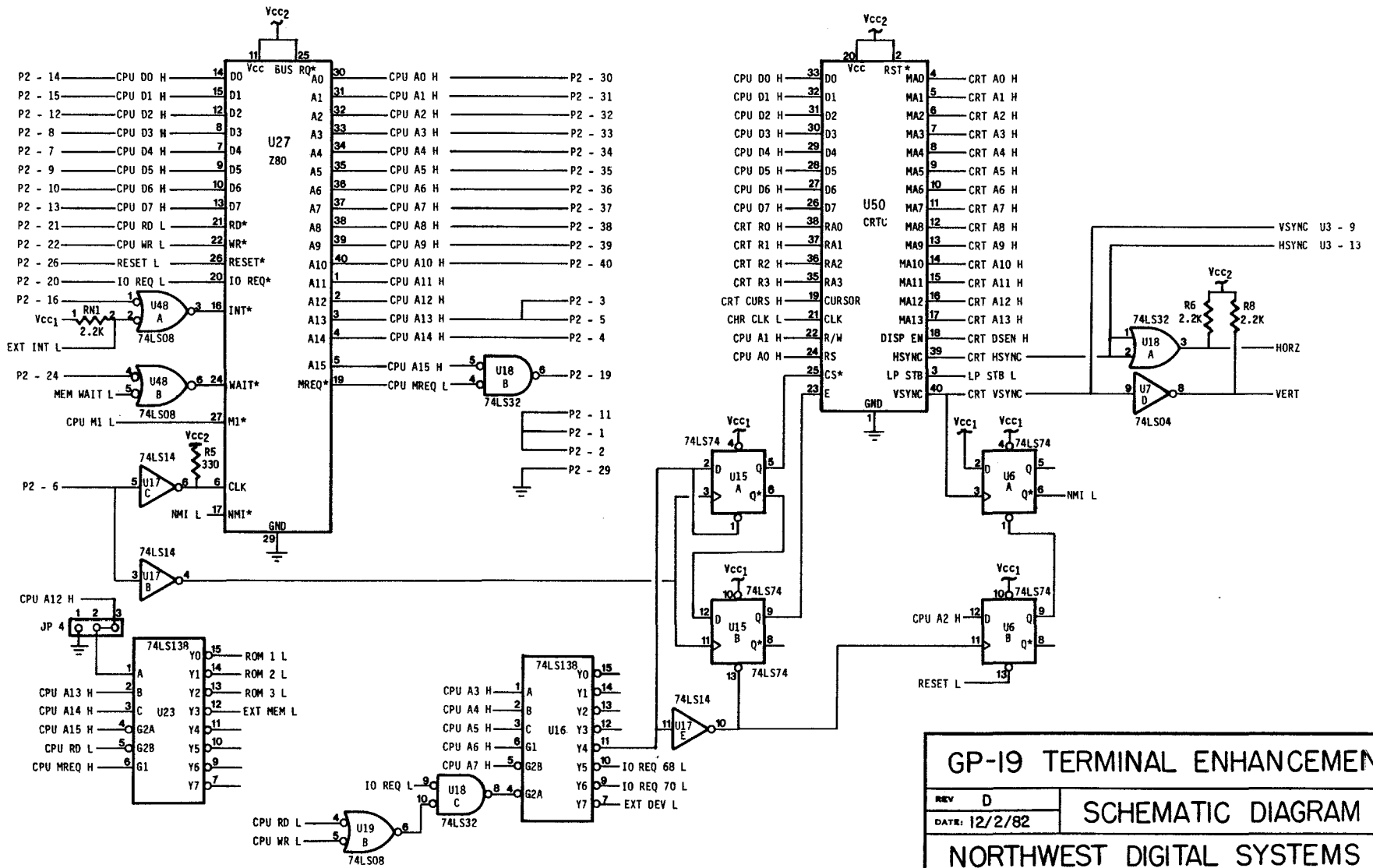
```

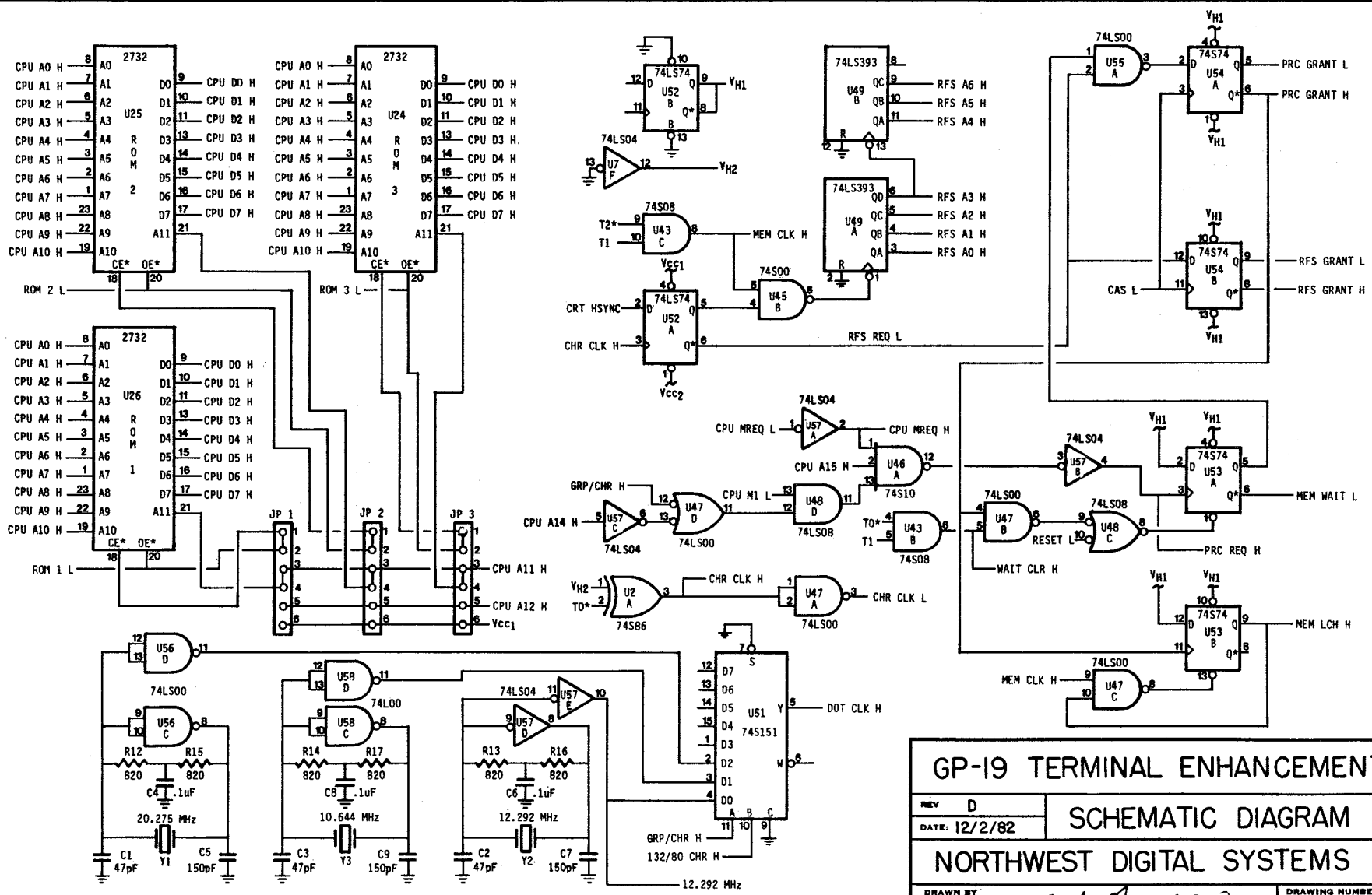
SECRET  
COMMUNICATIONS SECURITY ESTABLISHMENT  
WASHINGTON, D.C. 20340  
FORM 104-100 (REV. 10-65)  
104-100-100

### APPENDIX H

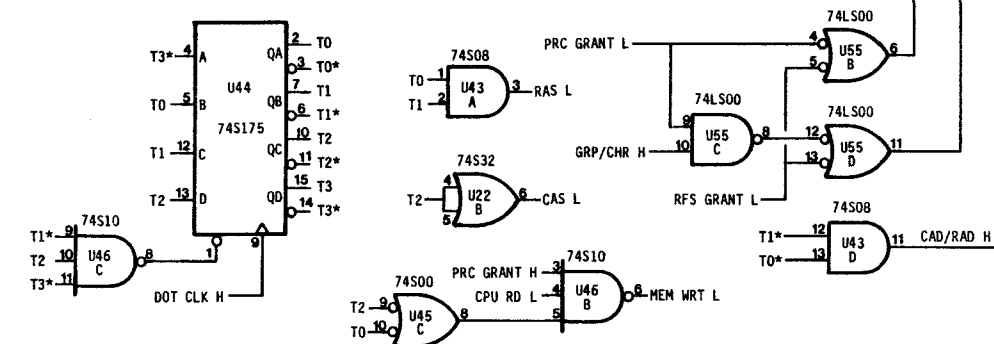
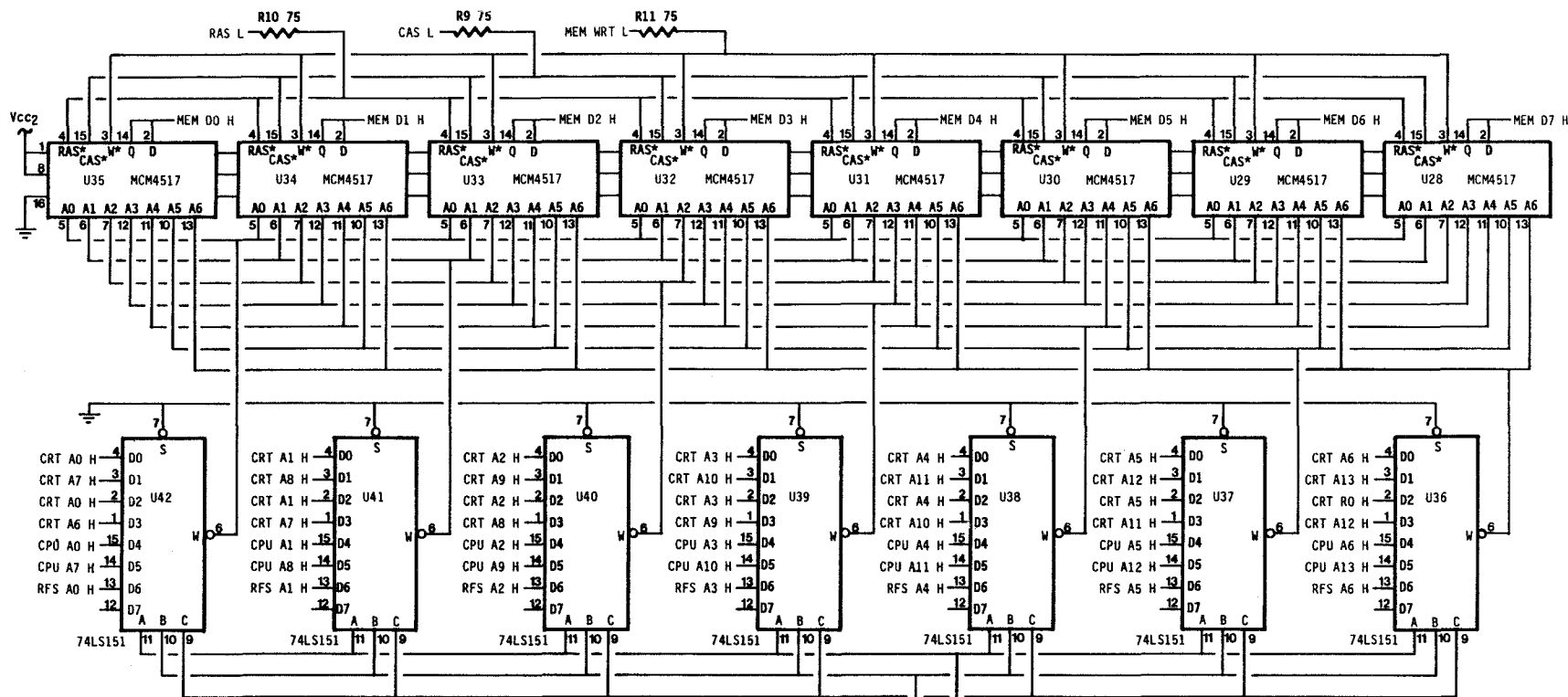
### CIRCUIT DIAGRAMS







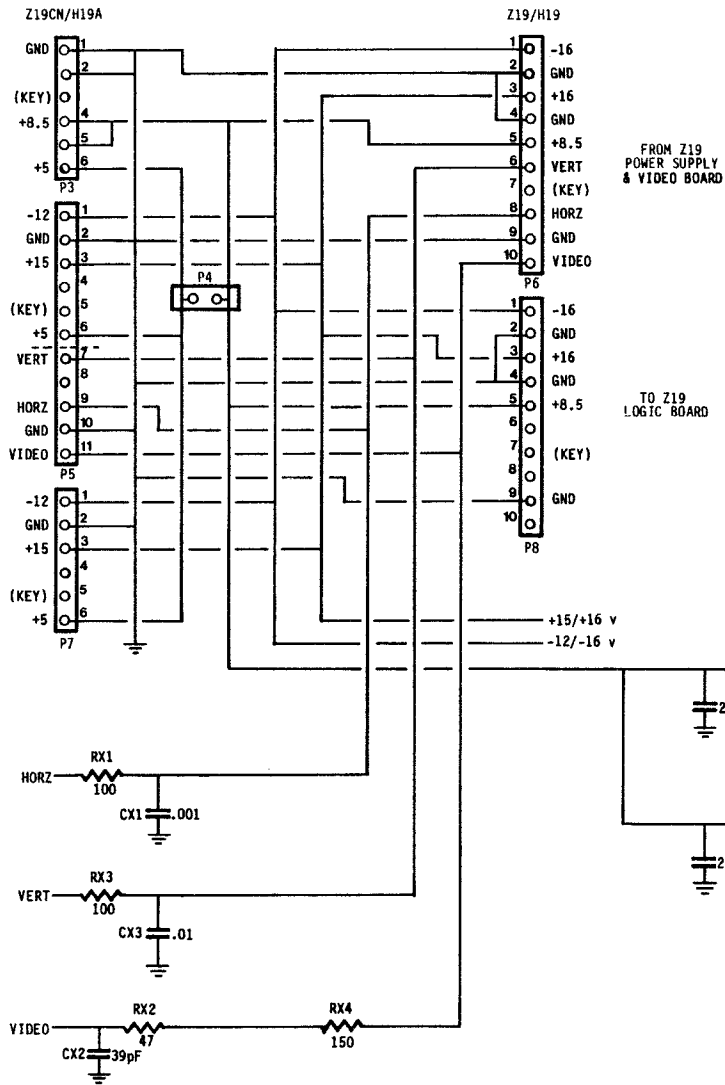
|                                      |                          |
|--------------------------------------|--------------------------|
| <b>GP-19 TERMINAL ENHANCEMENT</b>    |                          |
| REV D                                | <b>SCHEMATIC DIAGRAM</b> |
| DATE: 12/2/82                        |                          |
| <b>NORTHWEST DIGITAL SYSTEMS</b>     |                          |
| DRAWN BY <i>John H. ... B. Davis</i> |                          |
| DRAWING NUMBER 12-00001-03           |                          |



|                                   |                               |
|-----------------------------------|-------------------------------|
| <b>GP-19 TERMINAL ENHANCEMENT</b> |                               |
| REV D                             | <b>SCHEMATIC DIAGRAM</b>      |
| DATE: 12/2/82                     |                               |
| <b>NORTHWEST DIGITAL SYSTEMS</b>  |                               |
| DRAWN BY<br><i>John H. ...</i>    | DRAWING NUMBER<br>12-00001-04 |

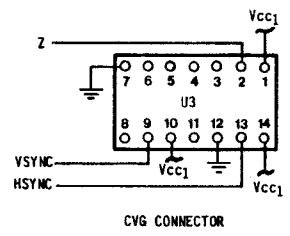




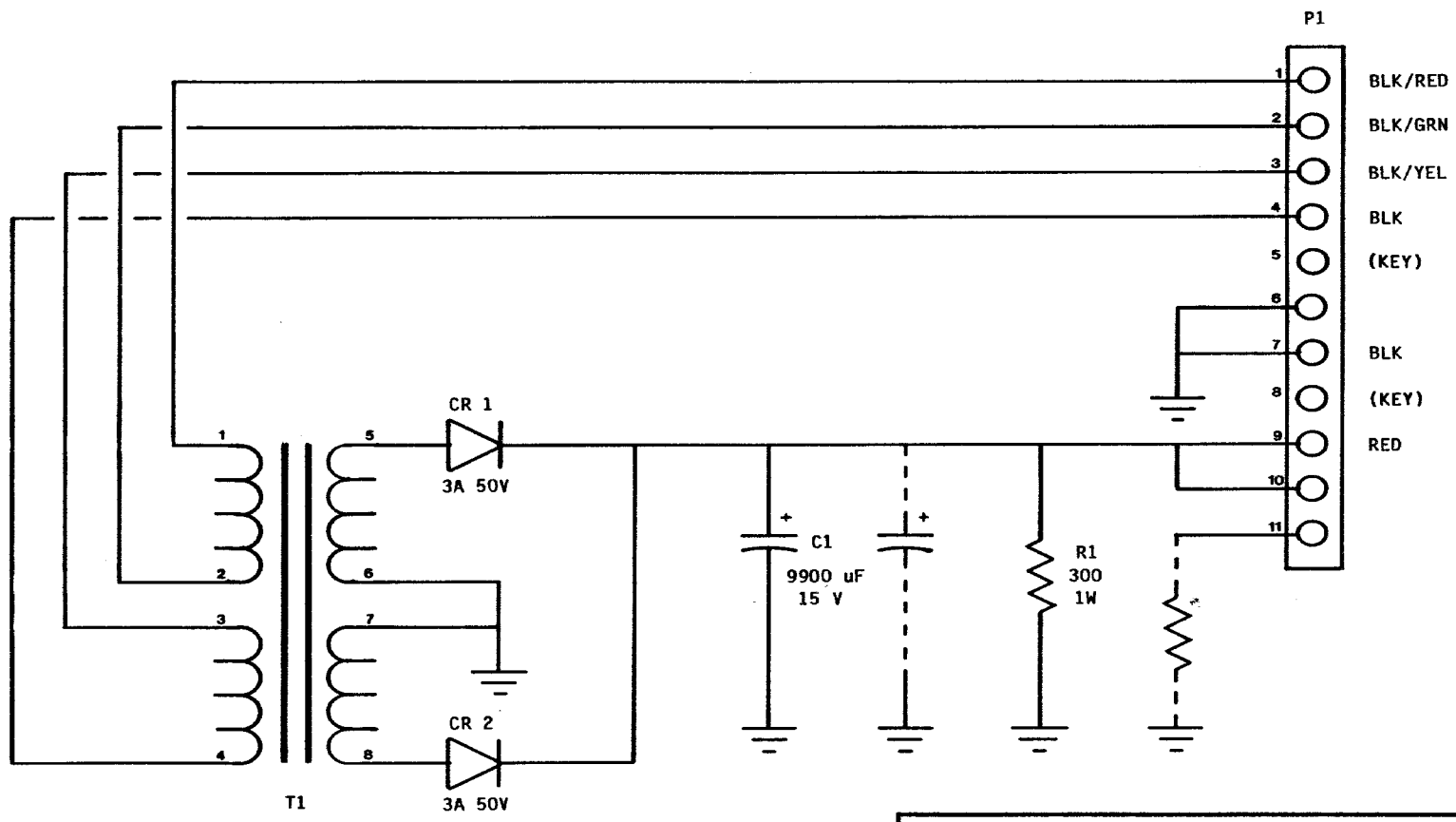


I/O CONNECTOR

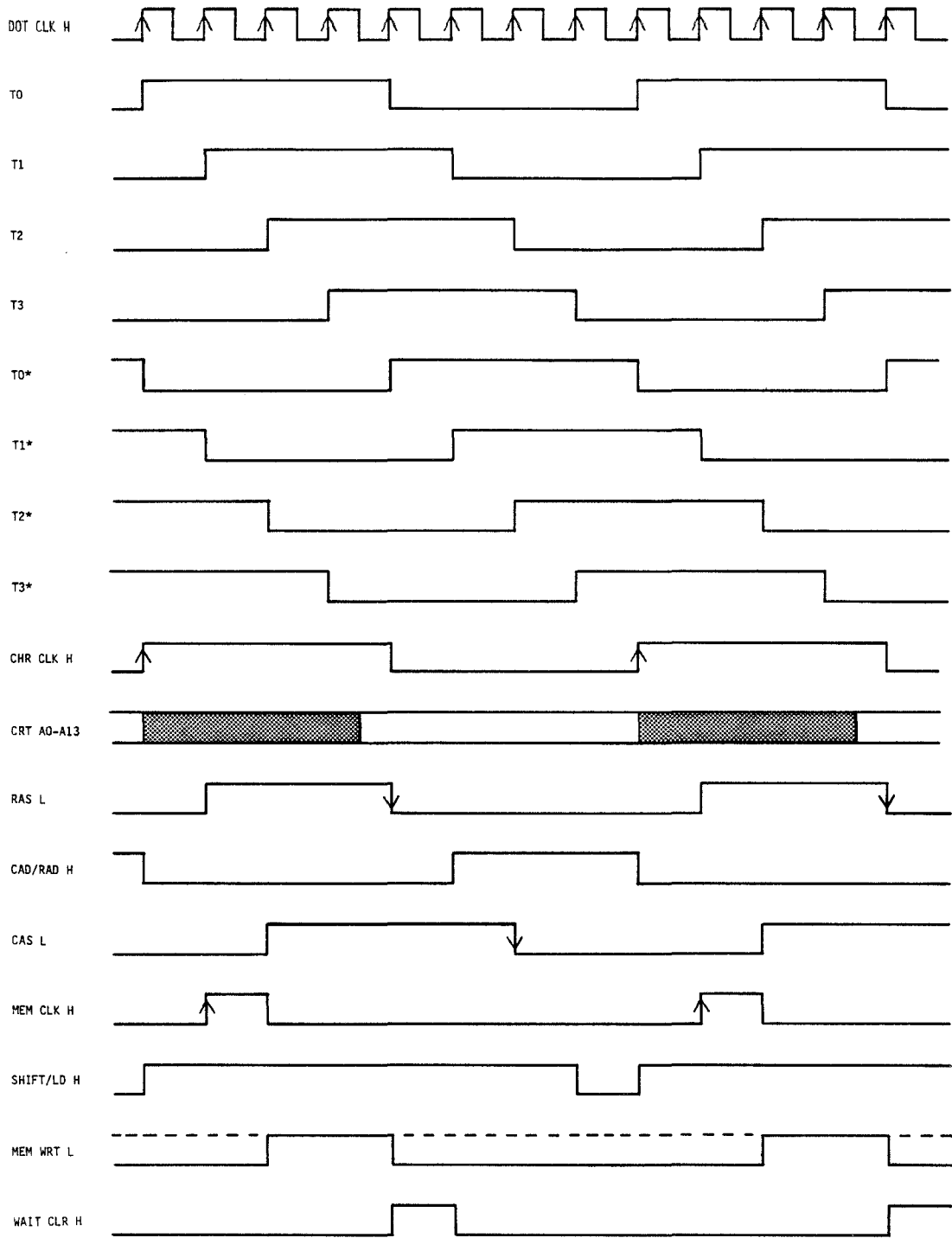
|            |    |    |            |
|------------|----|----|------------|
| GND        | 40 | 39 | Vcc (+5v)  |
| CPU A14 H  | 38 | 37 | CPU A13 H  |
| CPU D0 H   | 36 | 35 | CPU A0 H   |
| CPU D1 H   | 34 | 33 | CPU A1 H   |
| CPU D2 H   | 32 | 31 | CPU A2 H   |
| CPU D3 H   | 30 | 29 | CPU A3 H   |
| CPU D4 H   | 28 | 27 | CPU A4 H   |
| CPU D5 H   | 26 | 25 | CPU A5 H   |
| CPU D6 H   | 24 | 23 | CPU A6 H   |
| CPU D7 H   | 22 | 21 | CPU A7 H   |
| CPU RD L   | 20 | 19 | CPU A8 H   |
| CPU WR L   | 18 | 17 | CPU A9 H   |
| EXT DEV L  | 16 | 15 | CPU A10 H  |
| RESET L    | 14 | 13 | CPU A11 H  |
| EXT INT L  | 12 | 11 | CPU A12 H  |
| LP STB L   | 10 | 9  | EXT MEM L  |
| CPU MREQ L | 8  | 7  | 12.292 MHz |
| -12/-16 v  | 6  | 5  | +15/+16 v  |
| CPU A15 H  | 4  | 3  | NC         |
| GND        | 2  | 1  | GND        |



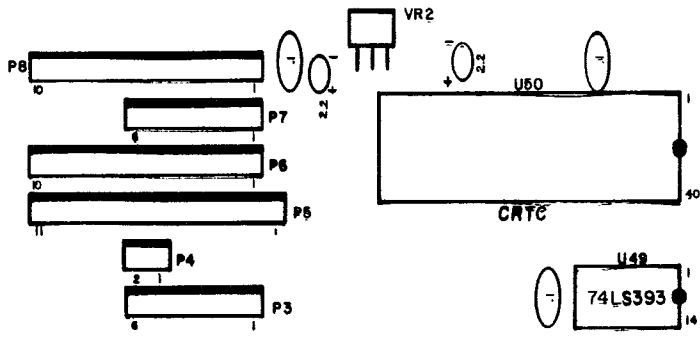
|                                |                               |
|--------------------------------|-------------------------------|
| GP-19 TERMINAL ENHANCEMENT     |                               |
| REV D                          | SCHEMATIC DIAGRAM             |
| DATE: 12/2/82                  |                               |
| NORTHWEST DIGITAL SYSTEMS      |                               |
| DRAWN BY<br><i>John H. ...</i> | DRAWING NUMBER<br>12-00001-06 |



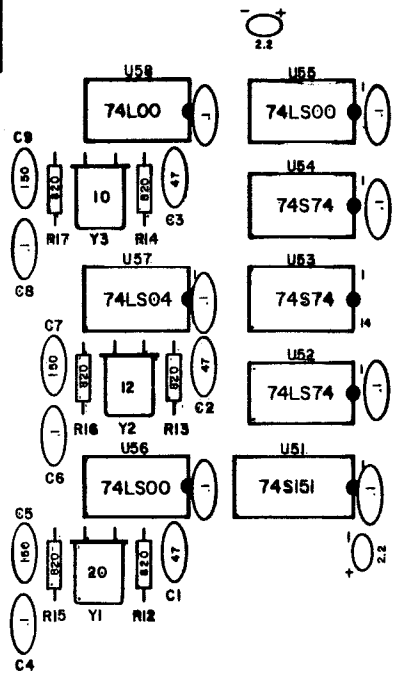
|                                   |                                  |
|-----------------------------------|----------------------------------|
| <b>GP-19 TERMINAL ENHANCEMENT</b> |                                  |
| REV D                             | -CN POWER SUPPLY                 |
| DATE: 12/2/82                     | <b>NORTHWEST DIGITAL SYSTEMS</b> |
| DRAWN BY<br><i>John P. ...</i>    | DRAWING NUMBER<br>12-00001-07    |



|                               |                               |
|-------------------------------|-------------------------------|
| GP-19 TERMINAL ENHANCEMENT    |                               |
| REV D                         | TIMING DIAGRAM                |
| DATE: 12/2/82                 |                               |
| NORTHWEST DIGITAL SYSTEMS     |                               |
| DRAWN BY<br><i>Jan H. ...</i> | DRAWING NUMBER<br>12-00001-01 |



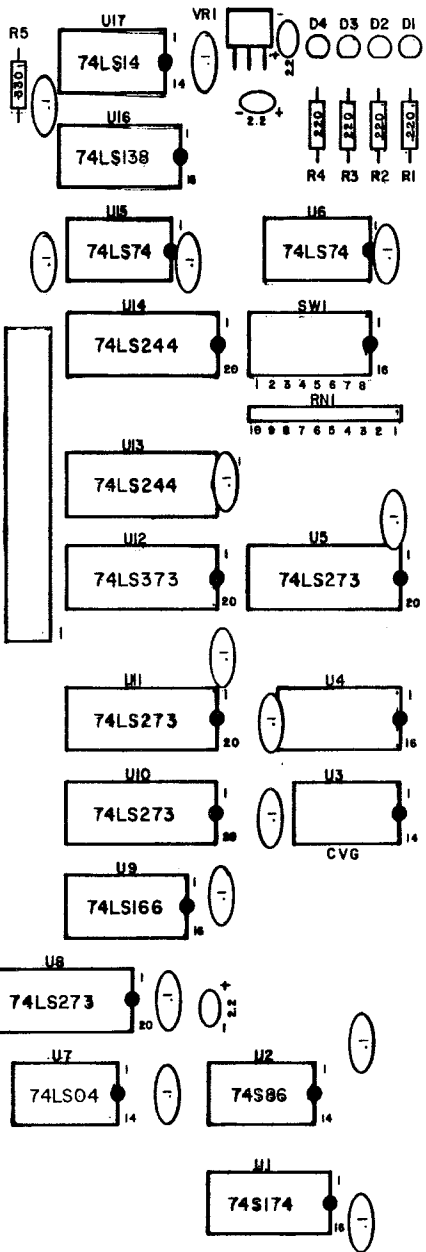
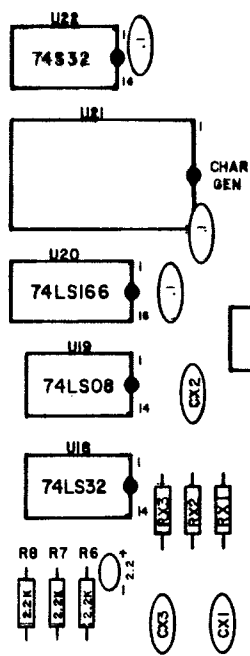
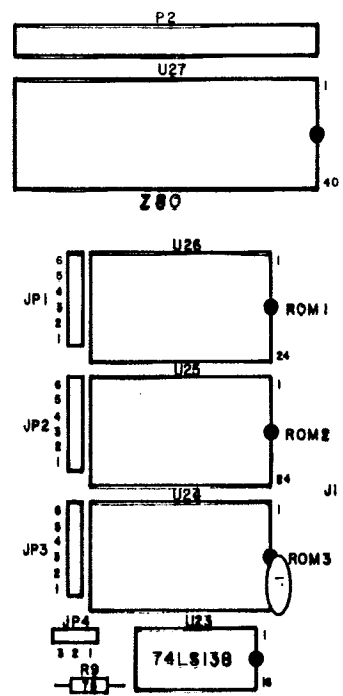
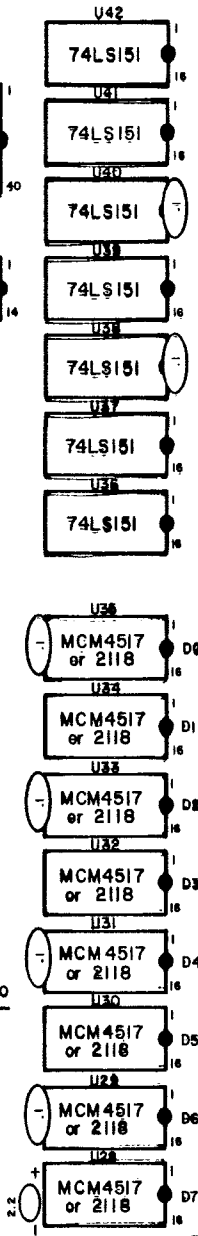
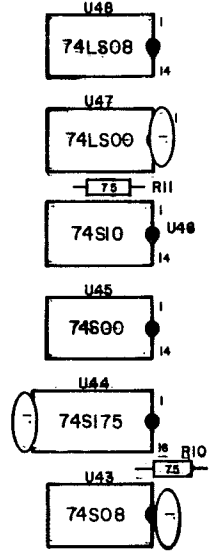
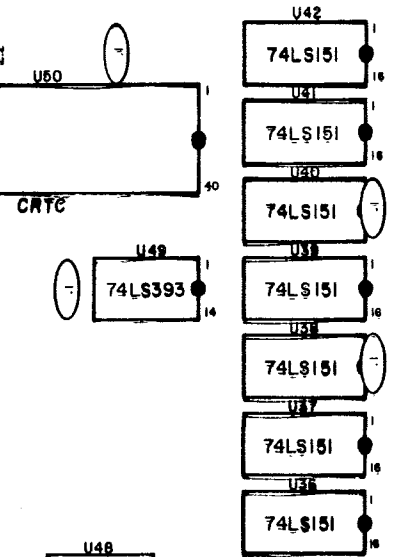
SER. NO.



NDS GP19

© 8/82

NORTHWEST DIGITAL SYSTEMS  
SEATTLE, WASHINGTON



PCB ASSY 09-00001-02

APPENDIX I

PARTS LIST

| <u>Component ID</u> | <u>NDS Part #</u> | <u>Item</u> | <u>Description</u>          |
|---------------------|-------------------|-------------|-----------------------------|
| U1                  | 30-74174-01       | 74S174      | Hex D-Type Flip-Flops       |
| U2                  | 30-07486-01       | 74S86       | Quad 2-input EX-OR          |
| U5                  | 30-74273-02       | 74LS273     | Octal D-Type Flip-Flops     |
| U6                  | 30-07474-02       | 74LS74      | Dual D-Type Flip-Flops      |
| U7                  | 30-07404-02       | 74LS04      | Hex Inverters               |
| U8                  | 30-74273-02       | 74LS273     | Octal D-Type Flip-Flops     |
| U9                  | 30-74166-02       | 74LS166     | 8-Bit Shift Register        |
| U10 - U11           | 30-74273-02       | 74LS273     | Octal D-Type Flip-Flops     |
| U12                 | 30-74373-02       | 74LS373     | Octal D-Type Latches        |
| U13 - U14           | 30-74244-02       | 74LS244     | Octal Buffer                |
| U15                 | 30-07474-02       | 74LS74      | Dual D-Type Flip-Flops      |
| U16                 | 30-74138-02       | 74LS138     | 3-To-8 Line Decoder         |
| U17                 | 30-07414-02       | 74LS14      | Hex Schmitt Inverters       |
| U18                 | 30-07432-04       | 74ALS32     | Quad 2-input OR             |
| U19                 | 30-07408-02       | 74LS08      | Quad 2-input AND            |
| U20                 | 30-74166-02       | 74LS166     | 8-Bit Shift Register        |
| U21                 | 36-00001-01       | 2732A-3     | 32K EPROM (350 ns) CG-1     |
| U22                 | 30-07432-01       | 74S32       | Quad 2-input OR             |
| U23                 | 30-74138-02       | 74LS138     | 3-To-8 Line Decoder         |
| U24                 | 36-00004-01       | 2732        | 32K 450 ns EPROM ROM 3      |
| U25                 | 36-00003-01       | 2732        | 32K 450 ns EPROM ROM 2      |
| U26                 | 36-00002-01       | 2732        | 32K 450 ns EPROM ROM 1      |
| U27                 | 32-00001-01       | Z80         | 2 MHz Z80 CPU               |
| U28 - U35           | 32-00003-01       | 2118-7      | 16K 5V Dynamic RAM (150 ns) |
|                     | or 32-00004-01    | MCM4517-P20 | 16K 5V Dynamic RAM (150 ns) |
| U36 - U42           | 30-74151-02       | 74LS151     | 1-Of-8 Data Selector        |
| U43                 | 30-07408-01       | 74S08       | Quad 2-input AND            |
| U44                 | 30-74175-01       | 74S175      | Quad D-Type Flip-Flops      |
| U45                 | 30-07400-01       | 74S00       | Quad 2-input NAND           |
| U46                 | 30-07410-04       | 74ALS10     | Triple 3-input NAND         |
| U47                 | 30-07400-02       | 74LS00      | Quad 2-input NAND           |
| U48                 | 30-07408-02       | 74LS08      | Quad 2-input AND            |
| U49                 | 30-74393-02       | 74LS393     | Dual 4-Bit Binary Counters  |

## PARTS LIST

| <u>Component ID</u> | <u>NDS Part #</u> | <u>Item</u> | <u>Description</u>       |
|---------------------|-------------------|-------------|--------------------------|
| U50                 | 32-00002-01       | HD46505SP   | CRT Controller           |
| U51                 | 32-74151-01       | 74S151      | 1-Of-8 Data Selector     |
| U52                 | 32-07474-02       | 74LS74      | Dual D-Type Flip-Flops   |
| U53 - U54           | 32-07474-01       | 74S74       | Dual D-Type Flip-Flops   |
| U55 - U56           | 32-07400-02       | 74LS00      | Quad 2-input NAND        |
| U57                 | 32-07404-02       | 74LS04      | Hex Inverters            |
| U58                 | 32-07400-03       | 74L00       | Quad 2-input NAND        |
| R1 - R4             | 40-00221-01       | Resistor    | 220 ohm 250 mW 5%        |
| R5                  | 40-00331-01       | Resistor    | 330 ohm 250 mW 5%        |
| R6 - R8             | 40-00222-01       | Resistor    | 2.2K ohm 250 mW 5%       |
| R9 - R11            | 40-00750-01       | Resistor    | 75 ohm 250 mW 5%         |
| R12 - R17           | 40-00821-01       | Resistor    | 820 ohm 250 mW 5%        |
| RN1                 | 40-00222-02       | Resistor    | 2.2K ohm SIP Network     |
| * RX4               | 40-00151-01       | Resistor    | 150 ohm 250 mW 5%        |
| D1 - D4             | 48-00001-01       | LED         |                          |
| SW1                 | 52-00001-01       | Dip Switch  | 8 Position               |
| Y1                  | 50-00003-01       | Crystal     | 20.282 MHz               |
| Y2                  | 50-00002-01       | Crystal     | 12.292 MHz               |
| Y3                  | 50-00001-01       | Crystal     | 10.644 MHz               |
| C1,C2,C3            | 42-00003-01       | Capacitor   | Ceramic, 47 pF           |
| C4,C6,C8            | 42-00001-01       | Capacitor   | Ceramic, 0.1 uF 12V      |
| C5,C7,C9            | 42-00004-01       | Capacitor   | Ceramic, 150 pF          |
| VR1 - VR2           | 34-00001-01       | 7805CT      | +5 Volt, 1 Amp Regulator |
| 0.1 uF Cap          | 42-00001-01       | Capacitor   | 0.1 uF 12V Ceramic       |
| 2.2 uF Cap          | 42-00002-01       | Capacitor   | 2.2 uF 25V Tantalum      |
| Cable Assy          | 62-00001-01       | Cable       | Processor Interconnect   |
| * Cable Assy        | 62-00002-01       | Cable       | Power Interconnect       |
| % Cable Assy        | 62-00003-01       | Cable       | Power Interconnect       |
| # Cable Assy        | 62-00004-01       | Cable       | Power Supply             |
| Label               | 80-00001-01       |             | Keyboard Label           |
| Manual              | 14-00001-01       |             | Instruction Manual       |

# - denotes parts only for Z19-CN Terminals

\* - denotes parts only for Z19 or H19 Terminals

% - denotes parts only for Z19-CN or H19A Terminals

## PARTS LIST

| <u>Component ID</u>   | <u>NDS Part #</u> | <u>Item</u> | <u>Description</u>       |
|-----------------------|-------------------|-------------|--------------------------|
| # +5V Supply          | 82-00001-01       | Pwr Supply  | Unreg. +5 Volt 1.2 Amp   |
| 82-00001-01 Includes: |                   |             |                          |
| T1                    | 53-00001-01       | Transformer | Signal DST-6-16          |
| C1                    | 42-00005-01       | Capacitor   | 9900 uF 15V Electrolytic |
| D1,D2                 | 48-00002-01       | Diode       | 3 Amp 50 Volt            |
| R1                    | 40-00301-04       | Resistor    | 300 ohm 1W 5%            |
| Screw                 | 60-01004-01       | #12 X .500  | Self-Tapping Washer Head |
| In-Line Tap           | 64-00005-01       | Scotchlok   | #558 Tap Connector       |

# - denotes parts only for Z19-CN Terminals