

AED 512/767
COLOR GRAPHICS TERMINALS
USERS MANUAL



ADVANCED ELECTRONICS DESIGN, INC.

**440 POTRERO AVENUE
SUNNYVALE, CA 94086
(408) 733-3555**

APRIL, 1983

PART NO. 990003-01 REV-A

TABLE OF CONTENTS

1. INTRODUCTION/OVERVIEW (Page 1-1)

- 1.1 Applications and Features (1-2)
- 1.2 System Architecture (1-3)
- 1.3 Anti-Aliased Vectors (1-12)
- 1.4 Blue Line Grid (1-12)

2. SPECIFICATIONS (Page 2-1)

- 2.1 Terminal Specifications (2-1)
- 2.2 Monitor Specifications (2-10)
- 2.3 DEC Compatible Interfaces (2-13)

3. INSTALLATION (Page 3-1)

- 3.1 Unpacking (3-1)
- 3.2 Inspection (3-1)
- 3.3 Electrical Power-up (3-1)
- 3.4 Verification of Operation (3-8)
- 3.5 Communications (3-12)
- 3.6 Software (3-12)
- 3.7 Overnight or Continuous Use (3-12)
- 3.8 Switch Settings (3-12)
- 3.9 Installation Check List (3-17)

4. KEYS AND INDICATORS (Page 4-1)

- 4.1 Special Function Keys (4-1)
- 4.2 Standard Function Keys (4-8)
- 4.3 Indicators (4-8)
- 4.4 The Normal (7 x 12) Font (4-10)
- ASCII Character Code Charts

5. TERMINAL COMMAND PROTOCOL (TCP) (Page 5-1)

- 5.1 Introduction (5-1)
- 5.2 Functional Index of Commands (5-2)
- 5.3 Alphabetical Ordering of (5-4)
Commands

REGARDING PRODUCTION OF THIS AED 512/767 GRAPHICS/IMAGING TERMINAL USERS MANUAL

Chapters 1 through 4 have been phototypeset for appearance and readability. Chapter 5, the Terminal Command Protocol, has not been produced in this way because it is expected that continuing improvements in firmware will force its frequent revision.

Please direct your comments and suggestions regarding this manual to AED Technical Publications Department.

The responsibility for technical accuracy of this manual, assured by their final review, is borne by the AED Engineering Department.

TRADEMARK/COPYRIGHT NOTICE

This manual has been prepared by Advanced Electronics Design, Inc. for use by its licensees, customers, representatives and employees. AED reserves the right to make changes to this manual, or to the products described, at any time and without prior notice. No reproduction of this manual shall be done, in whole or in part, by any method, without prior written approval of AED.

While considerable care has been exercised, AED does not assume responsibility for errors in this manual and shall not be responsible for damages, direct or consequential, arising from its use.

Copyright Pending
ADVANCED ELECTRONICS DESIGN, INC.
All rights reserved
Printed in the United States of America

SUPEROAM is a registered trademark of Advanced Electronics Design, Inc.

TEKTRONIX and **PLOT 10** are registered trademarks of Tektronix, Inc.

DEC, PDP, LSI-11, UNIBUS and **Q-BUS** are registered trademarks of Digital Equipment Corporation.

CONGRATULATIONS ON YOUR PURCHASE OF AN AED COLOR GRAPHICS TERMINAL!

Upon receipt of the information requested below, you automatically become a member of our GRAPHICS USERS' GROUP which was formed at ACM SIGGRAPH in 1981.

Please forward this sheet to:

AED USERS' GROUP
c/o AED, Inc.
440 Potrero Avenue
Sunnyvale, CA 94086 USA

The information will be compiled into our data base and you will receive the latest newsletter and any additional mailings from the Group.

If there is more than one user for this terminal, or additional users at your facility, please have them supply this information also and forward it to AED.

Again, Congratulations, and Welcome to the group.

Type of terminal: _____ AED 512 _____ AED 767 _____ AEDS 11

NAME: _____ DATE: _____

TITLE: _____

COMPANY: _____

ADDRESS: _____

MAIL STOP/P.O.BOX: _____

CITY, STATE, ZIP: _____

TELEPHONE: () _____ EXT. _____

TELEX/TWX: _____

ADVANCED ELECTRONICS DESIGN, INC.

440 Potrero Avenue
Sunnyvale, CA 94086

(408) 733-3555
Telex 357498

INSPECTION AND INSTALLATION REPORT

Please help us to help you by providing this feedback on the condition of our equipment at arrival and its performance during initial operation. Mark the boxes as applicable and use the comments section to tell us of anything unusual that is necessary to bring the equipment on-line.

Thank you!

Quality Assurance Department

	YES	NO
1. Any visible damage to shipping carton?	___	___
2. Any damage to unit visible after unpacking?	___	___
3. Were necessary manuals, documents included?	___	___
4. Were all cables, accessories delivered or explained?	___	___
5. Were printed circuit boards in properly?	___	___
6. Would unit power up initially? If not, explain below.	___	___
7. Would unit go on line and operate properly? If "NO" please indicate below what adjustments were made.	___	___

COMMENTS: (Use extra sheets, if necessary) _____

Serial Number: _____ Location: _____

Product/Model No.: _____ Customer: _____

Form Submitted By: _____ Date: _____
(Name)

PLEASE FOLD AND MAIL TO THE ADDRESS ON REVERSE

AFFIX
POSTAGE

TO:

**ADVANCED ELECTRONICS DESIGN, INC.
440 POTRERO AVENUE
SUNNYVALE, CA 94086**

FOLD

FOLD

1. INTRODUCTION/OVERVIEW

1.1 APPLICATIONS AND FEATURES (1-2)

1.2 SYSTEMS ARCHITECTURE (1-3)

- 1.2.1 Communications (1-3)
- 1.2.2 Video Storage (1-5)
- 1.2.3 Zooming The Display (1-7)
- 1.2.4 Panning the Display (1-7)
- 1.2.5 Monochrome (Black and White) Output (1-8)
- 1.2.6 Joystick Input (1-8)
- 1.2.7 Tracing The Signal Flow (1-8)
- 1.2.8 Superoam (1-10)

1.3 ANTI-ALIASED VECTORS (1-12)

1.4 BLUE LINE GRID (1-12)

1 INTRODUCTION/OVERVIEW

This manual is written for users of both the AED 767 and the AED 512. Sections pertaining only to the AED 767 are written on a grey background; when material applies only to the AED 512, it will be stated so explicitly; otherwise, whatever is written applies to both the AED 512 and the AED 767. In this chapter, the system architecture is discussed from a logical or signal flow point of view. An overview of zoom, pan, and other special system features is provided.

Chapter 2 discusses the *specifications* on performance of the terminal, its interface with a host, and the monitor.

Chapter 3 contains detailed directions for *unpacking*, for physically *inspecting*, and for *installing* a new terminal and monitor. The chapter ends by giving some simple tests designed to help verify that everything is working as intended.

Chapter 4 describes the *keys* and *indicator lights* which are needed for interactive graphics and for programming the terminal in the local mode. This chapter also contains a table of ASCII character codes.

Chapter 5 treats in detail each of the Terminal Command Protocol (TCP) *commands*; there are approximately 100 of them. Many of the commands are illustrated by examples.

1.1 APPLICATIONS AND FEATURES

The **AED Color Graphics/Imaging Terminal** is a powerful tool for displaying data in a variety of applications. This chapter will provide an introduction to, and overview of, the terminal features, system architecture and theory of operation.

One large class of applications uses the terminal as a computer peripheral for display and modification of images. High speed development of complex multi-colored images is made possible by transfer of data from a host computer through the direct video memory access (DVMA) port in the terminal.

Another large class of applications uses the AED 512 or AED 767 as a telecommunications terminal connected to a remote host through an RS232-C interface. Keyboard and joystick inputs are integral to the terminal, and a graphics tablet is readily interfaced and commonly used.

Outstanding features of the **AED 512/767** are summarized below:

- ▶ Wide variety of colors and intensities available
- ▶ Horizontal and vertical scrolling
- ▶ Independent, wide range zoom on horizontal and vertical axes
- ▶ Unrestricted polygon (closed curve) fill
- ▶ Superoam® (panning over images significantly larger than full screen size), firmware and software selectable. (512 only).
- ▶ Single pixel addressability via host computer, keyboard, joystick, or tablet.
- ▶ Three selectable cursor shapes; blink colors selectable among any eight sets of two colors - blink rate also selectable
- ▶ **Blue line grid (767 only)**
- ▶ **Anti-Aliased vectors (767 only)**
- ▶ **Flicker free operation (767 only)**
- ▶ Serial and parallel data transfers, direct memory access, and run-length encoding for programming efficiency.
- ▶ Hardware self-test
- ▶ Documented and warranted software
- ▶ Low cost, high performance

The desk-top **AED Color Graphics/Imaging Terminal** is comprised of two mechanical packages: the terminal and integral keyboard (base unit); and the monitor (upper unit). They are interconnected by cables carrying red, green, or blue video signals. The rack mountable terminal configuration has a separate, detachable keyboard.

The terminal operates in a *Local Mode* or in a *Remote Mode*. In the *Local Mode*, alphanumeric and graphic operations can be entered through the keyboard and the joystick. Depressing the key titled LOCAL, just after powering up or resetting the terminal, prepares the AED 512/767 to receive keyboard inputs. An underline alphanumeric cursor appears at the upper left-hand corner of the screen; its color is red on a black background. The characters, as typed, will appear on the screen in upper case (AED 512 default condition), each in a 7 x 12 dot matrix.

In the Lower Case Mode the operator may enter lower case characters and makes the keyboard operate like a familiar office typewriter. For example, a capital letter is obtained by depressing SHIFT and then striking the letter key. The default state of the AED 512 is UPPER CASE and the default state of the AED 767 is LOWER CASE.

If the operator depressed the ESC (Escape) key, the terminal lights the INTERPRETER indicator and enters the *Interpreter Mode*. In this mode, ASCII characters from the keyboard are interpreted as *command function codes* and *arguments*, performing functions on the screen consistent with the Terminal Command Protocol (single character alphas, binary numerics). The user may also change the encoding scheme to a "three character mnemonic" mode of operation. This is less efficient because of the greater number of key-strokes, but the commands are easier to remember. Several forms of arguments are selectable such as binary, decimal and hexadecimal.

The terminal may be connected to an external Host computer, either by a serial line or through the parallel DMA channel. Operating over a serial line, the terminal may be connected to a local computer or, via a modem, to a remote time-sharing computer. Communication rates range from 300 to 19,200 baud and higher. Other external equipment such as a digitizer tablet may be connected to the terminal through the Auxiliary RS232-C port.

1.2 SYSTEM ARCHITECTURE

1.2.1 Communications

Referring to the *system architecture* drawing (Figure 1.1), the Host and Auxiliary RS232-C *serial* ports are shown on the right hand portion of the drawing. Number 1 is the Host port; number 2 is the Auxiliary port. Note port number 1 has both a 20 milliamp current loop and an EIA (RS232-C) configuration; the auxiliary port has only an EIA configuration. The baud rates for both ports are independently settable a) from the Communication DIP switches on the rear of the terminal, b) from the keyboard in the Interpreter Mode, and c) from the host CPU under program control.

System Architecture

Exploded illustration shows the logic, display, and interface components of the AED512/767. Optional DMA interfaces for higher throughput are available from AED for a variety of popular mini/micro computers. All video outputs drive external monitors and video tape equipment, and the system's modular design permits the buyer to order the AED512/767 with or without the video monitor.

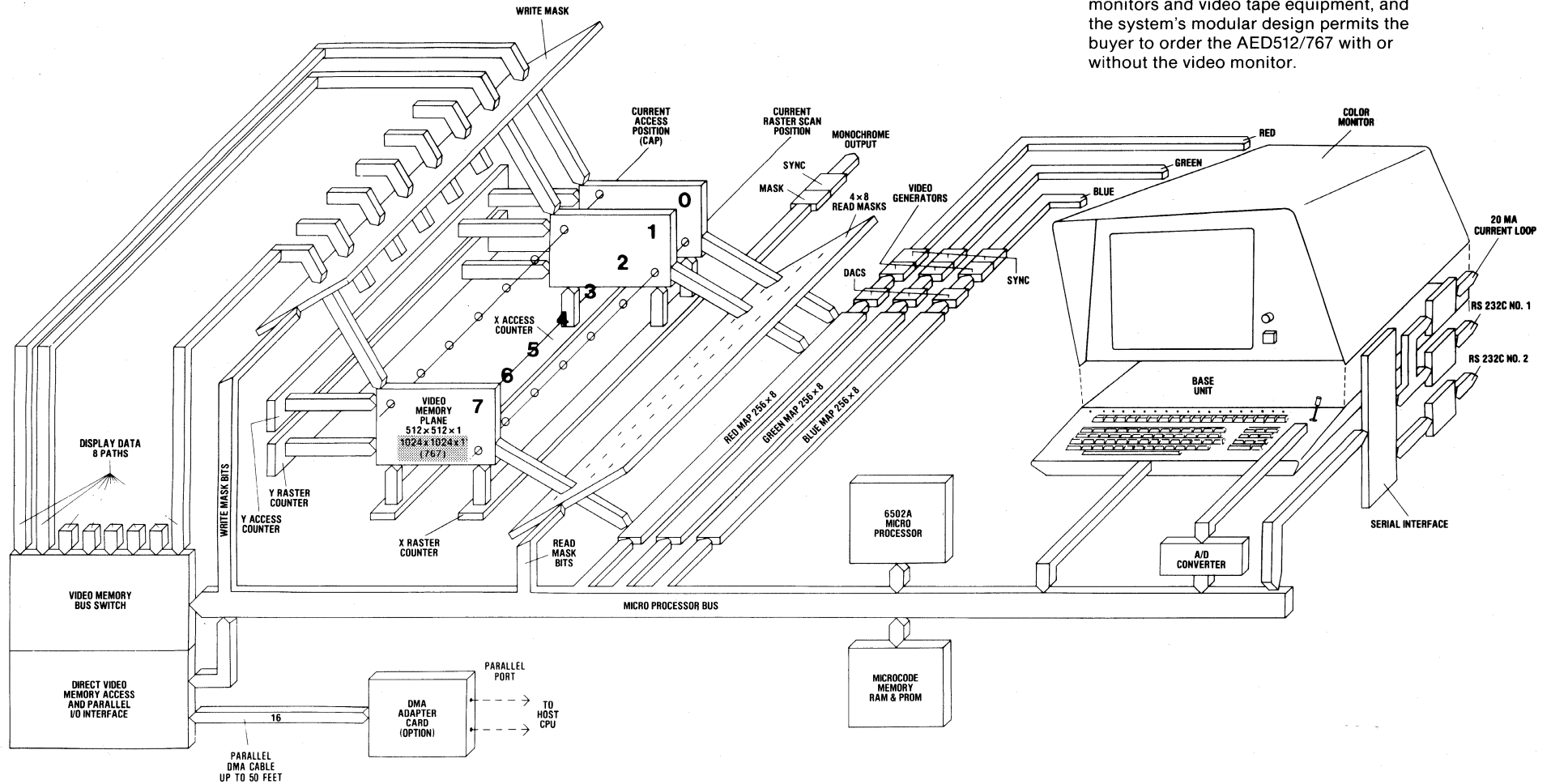


Figure 1.1
System Architecture

A second way for a computer to communicate with the AED 512/767 is through the *parallel* port shown in the lower left-hand portion of the system architecture drawing. This port transfers either 8 bit bytes or 16 bit words in parallel at a high rate. Imaging applications normally require parallel transfers because the screen is filled with pixels, each addressed from memory. This contrasts with graphics applications where programmed vectors and simplified conics are generated within the terminal and drawn on the video terminal screen. The parallel port requires a flat ribbon cable connected from the back of the terminal to the host adaptor card. AED has available parallel interface adaptor cards (also called Direct Memory Access-DMA-cards) for computers manufactured by Digital Equipment Corporation. Cards for other computers may be made available as required.

All data transfers and functions of the AED 512/767 are controlled by a 6502A microprocessor with the associated microcode memory (RAM and PROM). The 6502A has a cycle rate of two megahertz. PROMS are used for firmware while RAMS are used for temporary buffering and storage of user microcode, down-loadable from the host. The 6502A bus connector is available to the user. Considerable hardware circuitry is employed within the AED terminal to speed graphics generation and video refresh.

Various forms of communication between the terminal and the outside world are depicted in Figure 1.2.

1.2.2 Video Storage

Data entered through the channels mentioned above are transferred through the *video memory bus switch* and the *write mask* into the *video memory planes*. If all mask bits are set ("set" means "not masked") and all 8 video memory planes are installed, the 8 bits transferred through the write mask enter the memory planes. These **8 bits** provide a choice of "two to the eighth equals **256**" colors from a **palette of** "two to the twenty-fourth" or **16.8 million**. The *video screen* is refreshed from the contents of the video memory planes, through the read masks and the color lookup table (*red, green, and blue maps*). The digital signals are converted to analog by the *video DACs*, and the video synchronizing signal is added to all the video analog outputs. If all *read mask* bits are set, the contents of all 8 video memory planes are transferred into the color look-up table. Under microprocessor control, each of the color map signals (red, green, and blue) would earlier have been set to values in the range 0-255 decimal. The resultant analog video display signal is created by mapping of the video memory contents through the color map settings.

The write mask can be programmed to prohibit a particular bit or bits being transferred from the video memory bus switch to the video memory. There is one write mask bit per memory plane, i.e., the mask is 1 x 8 bits. Likewise, the read masks can be programmed to prohibit the output of the various video memory planes from addressing the color lookup table. These *write and read masks* enable the user to *selectively write into* and *read from* various *video memory planes*. One example is to have multiple images in sets of memory planes which are consecutively displayed on the video screen. This is one form of animation. If four images are to be sequentially displayed, then the first image can be written into memory planes 0 and 1, the second image into

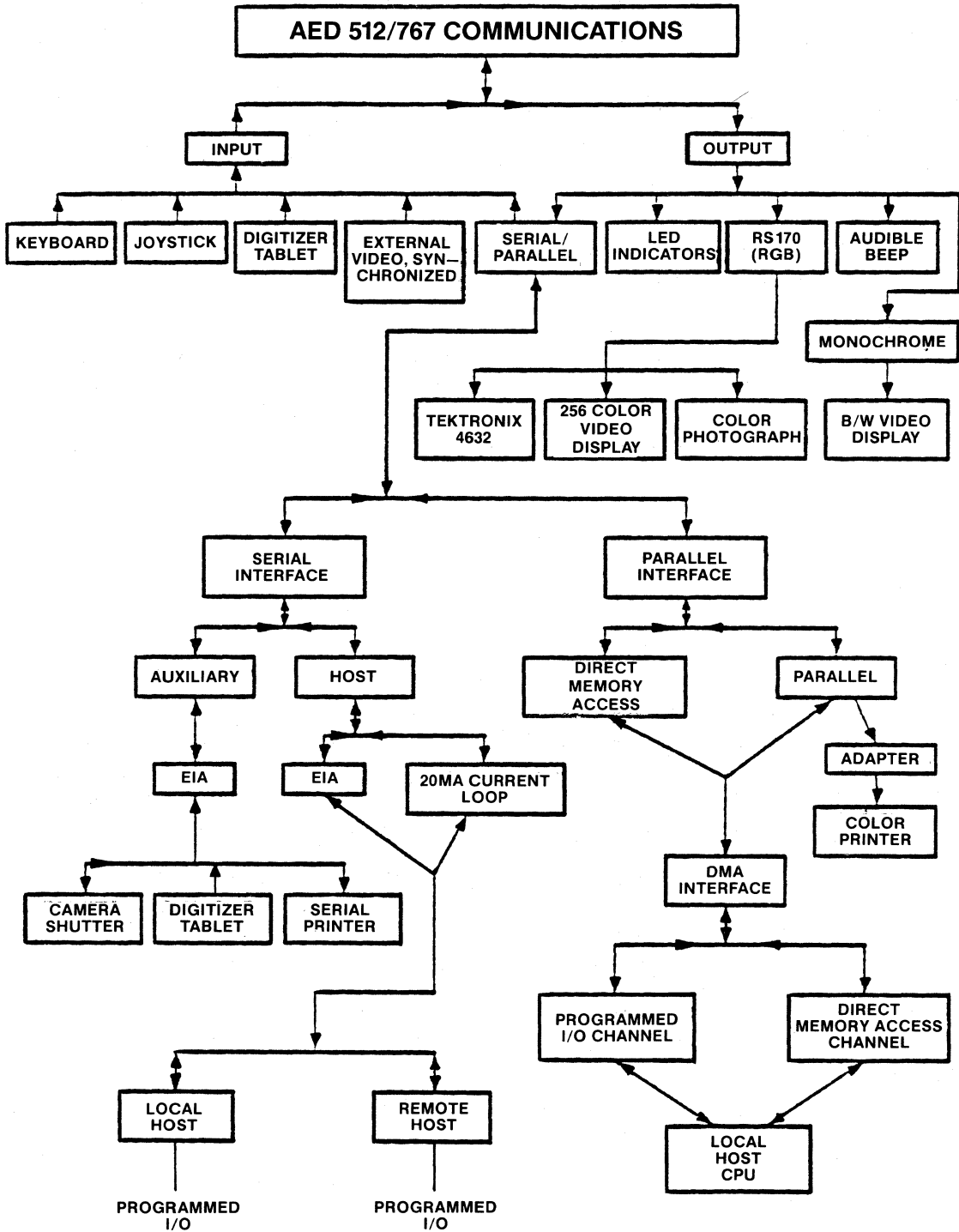


Figure 1.2
External Communications

memory planes 2 and 3, the third image into memories 4 and 5, and the fourth image into memory planes 6 and 7. By selecting memory planes 0 and 1 for a number of video frames, while masking out during the read cycle memory planes 2 through 7, only the first image will be shown on the screen. Next, memory planes 2 and 3 would be shown while memory planes 0 and 1, 4 and 5, and 6 and 7 are masked out, etc.

1.2.3 Zooming The Display

Zooming, or magnification of the display image, is accomplished by a process called *pixel replication*. At zoom X1, a pixel in video memory is equivalent to a single dot on the video screen and a dot consists of the usual microscopic red-green-blue triad employed in standard commercial TV's. When zooming, a pixel in the memory is visually extended both vertically and horizontally on the display so that a zoom X5 causes a pixel in the video memory to occupy a "5 dot by 5 line" area on the video screen. Likewise, a zoom X16 (the maximum available in the AED 512/767) has a "16 dot by 16 line" presentation for each pixel from video memory. Areas zoomed off the screen can be displayed by panning.

Additional flexibility is provided by the capability to *set the X and Y zoom factors independently*, under program control. Identical zoom levels in x and y are keyboard selectable by depressing special ZOOM IN and ZOOM OUT keys.

1.2.4 Panning the Display

Although the microprocessor is not fast enough to directly handle video data, the 6502A is used to set up the *raster counters* (hardware circuitry) which directly control access to the video memory planes. The contents of horizontal and vertical *origin registers* establish the upper left-most portion of the image on the video screen. Raster information is written to the screen from that position to higher order addresses in the video memory. Under zoom equal to 1, one scan line will include 512/640/768 pixels of information from the video memory planes. If the horizontal and vertical origins are set to 0, then the video screen will contain the full video image starting at the left most portion of the image. By enabling the *Pan Mode*, the image origin's position can be moved to other than the 0 video memory address location. Thus, the image can be panned (moved) from left to right under control of the joystick or by computer command. Please note, the AED 512 has 512 lines in memory but only 483 lines on the screen (RS-170 EIA Standard); therefore, 29 lines of video memory are not shown on the screen but can be panned or scrolled upwards or downwards for viewing. Likewise, the AED 767 has either 640H x 483V, 512H x 483V or 768H x 575V displayed from a 1024 x 1024 video memory.

1.2.5 Monochrome (Black and White) Output

A Monochrome output is available from *video memory plane 0*. This is useful should the operator wish to present text on a separate monochrome monitor but not have the text appear on the color display. It is easily achieved by connecting the monochrome NTSC or CCIR monitor to the RF connector labeled MONOCHROME on the back of the terminal, then employing the write masks to place text only in the memory plane 0 and not allowing it to pass on through to the color monitor. Please note that any scrolling, or panning, or zooming operation performed on the color display will also affect the monochrome output in the same manner.

1.2.6 Joystick Input

The joystick develops two analog voltages which are converted to digital by the *A-D converter*. By multiplexing, both X and Y position values are sent from the joystick to the *microprocessor bus* via the A-D converter. The joystick has two modes of operation - *position* and *rate*. In the *Position Mode*, there are standard and vernier scale factors which are selectable by one of the special function keys. In the standard/default scaling, the cross hair positioned by the cursor moves full scale for full mechanical deflection of the joystick. With vernier scaling, full scale deflection of the joystick moves the joystick 1/16th of the video screen width. In the *Rate Mode*, the cursor will be moving across the screen at a rate (speed) proportional to the joystick deflection magnitude, and at an angle determined by the angle of joystick deflection.

1.2.7 Tracing the Signal Flow

It may be instructive to trace the *signal flow* from the time a keystroke is entered from the keyboard or an ASCII character is placed on the serial line. The ASCII character enters the bus under microprocessor control and if the terminal is in the *Alpha Mode*, that character causes the microprocessor to jump to the suitable character generating PROM. This causes the appropriate bits to be transferred from the character generator to the video memory planes, thus establishing the icon at the proper place in video memory. Those bits are then transferred (through the color lookup table to the video screen) causing the character to appear at the correct position; the screen is refreshed so the character appears to be glowing continuously.

If the terminal is in the *Interpreter Mode*, then the ASCII character will cause the microprocessor to jump to a particular subroutine and execute the commands. For example, if the terminal is in the *Single Character, Non-Mnemonic, Binary Mode*, the ASCII character "O" is the command to draw a circle at the current access position (CAP). Immediately upon receipt of that character, the microprocessor will become BUSY as noted by the indicator above the special function keys. It will remain BUSY until a second character, such as A (equivalent to 65 decimal), is received for the radius (65 pixels) of the circle to be drawn. Upon receipt of the second character, the microprocessor will enter a subroutine for generating the pixels comprising the circle, and the circle will be drawn in the current color. If this sequence is followed by an ASCII I (Interior Fill command), the microprocessor will jump to that fill subroutine and cause the circle to be filled with the current color.

The following is a description of the events which take place within the AED 512/767 in the *Interpreter, Single Character, Non-Mnemonic, Binary Mode* of operation. The example used will be to draw a white vector from cursor location number 1 to cursor location number 2. to cursor location number 3... The following sequence of keys are depressed to achieve this drawing:

RESET	(Clears last pending command)
RESET	(Full RESET initializes terminal)
LOCAL	(Ignores external inputs from host)
ESC	(Escape causes terminal to enter Interpreter Mode)
CAPS LOCK	(AED 767 only, because default state is lower case)
C	(Change color)
CTRL G	(Color 7 decimal white)
CURSOR	(On) [Move joystick to position cursor at Location 1]
SHIFT SPACE	(Beginning point of Vector) [Move joystick to position cursor at Location 2]
SPACE	(Vector written to new cursor location) [Move joystick to position cursor at Location 3]
SPACE	(Vector written to new cursor location)
•	
•	

The first RESET cancels any previous pending keystroke action. The second RESET causes the terminal to go to the initializing mode which clears the screen, clears all temporary registers, conducts a check sum on the microprocessor memory, and posts the terminal signature, in red, in the upper left-hand corner of the screen. All operations including RESET are under control of the microprocessor. LOCAL causes the AED 512/767 to be isolated from any external host computer or auxiliary plotting device. ESC causes the terminal to go into the *Interpreter Mode*, such that any subsequent keystrokes will not be printed on the screen, but instead, will cause the microprocessor to enable various terminal operations. At this point, the INTERPRETER light will be lit. C is interpreted as a "change color" command to the microprocessor which causes the BUSY indicator to light. This sequence begins with C and must be followed by one more character designating the color to be selected; CONTROL G is equivalent to decimal 7 (from ACSII chart, Chapter 4), thereby selecting color number 7 which is white. At this point, the BUSY light extinguishes and the microprocessor is awaiting a new series of commands.

Depressing CURSOR informs the microprocessor that the user wishes to display the default cursor (green X) on the black (color 0) screen and have control of the position of that cursor with the *joystick* in the *Position Mode*. The cursor will now appear on the screen at some location depending upon the joystick deflection. *Manually moving the joystick will cause the cursor to move about on the screen.* This happens because the X and Y voltages developed by the joystick potentiometers are processed by the analog-to-digital converter, multiplexed, and then used by the microprocessor to control cursor position. The analog-to-digital converter is a dual-slope, integrating digital voltmeter type which accepts each analog voltage from the joystick potentiometers and converts it to a binary output. X and Y digital values are received by the microprocessor and used to draw the cursor in green into the video refresh memory. The *original contents of the video memory*, now occupied by the cursor symbol, *are temporarily stored in the microprocessor RAM* for subsequent replacement after the cursor is moved to a new location. The microprocessor also *changes the current access position (CAP) in video memory to the coordinates of the cursor center.* Manually moving the joystick to a new position causes the cursor to move to a new location on the screen. As the cursor sweeps across the screen, it is actually shifting through video memories; the pixels (bits) in video memory are temporarily stored in RAM, then replaced in video memory when the cursor moves to a new location.

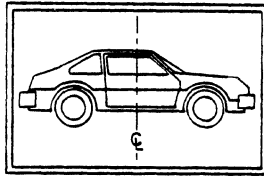
1.2.8 Superoam®

SUPEROAM is the trademarked name for the AED 512 terminal-unique feature that *allows the display of a software or keyboard selectable portion of an image larger than full screen size.* This is a very important attribute of the AED 512 which can be employed to advantage in computer aided design (CAD) applications. For example, a scaled replica of a B size drawing (11 x 17 inches) in an area measuring 512 pixels vertically and 1024 pixels horizontally can be written into eight memory planes. See *Figure 1.3.* The planes are configured into 2 adjacent stacks of 4 planes, each 512 x 512 for a total of 1024H x 512V pixels in area, with a total of 4 bits in depth (16 colors). By panning this image across the screen, 512 pixels horizontally, 483 lines vertically can be viewed as the "window" appears to move across the B size drawing. The *advantage* in having the 1024 pixels horizontally is to obtain *more resolution* than can be obtained with a single stack of 512 x 512 bit memories. However, since the 2 adjacent stacks of memory are only 4 bits deep, the number of colors available in this configuration is considerably less (16 versus 256) than if all 8 memory planes were stacked in one column. The programmer may utilize a *trade-off* among *image size, resolution, and number of available colors.*

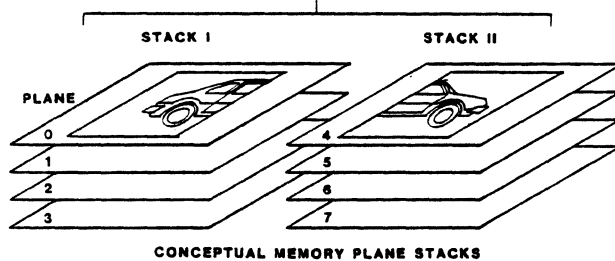
The implementation of this feature, SUPEROAM, is achieved by initially writing one 512 x 512 x 4 bit image in the left-most four memory planes (stack A). This can be achieved by using the write masks to allow the first image to appear only in memory planes 0 through 3. When the first 262,144 (i.e., 512 x 512) bits are written into memory planes 0 through 3, the write masks are altered to allow the next image to be written only into memory planes 4 through 7. At this point, *two completely separate images* representing a B size drawing are written into *Stack I* and *Stack II* memory planes.

SUPEROAM[®]

B SIZE DRAWING



VIDEO MEMORY



DISPLAY

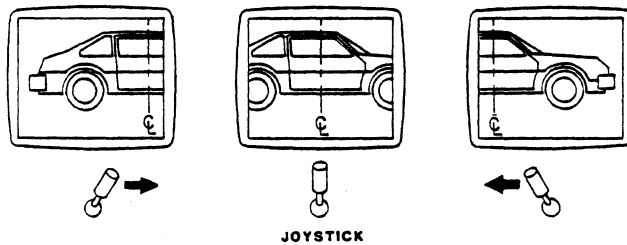


Figure 1.3 SUPEROAM[®]

The read masks are now used in the refresh (viewing) cycle to allow the B size drawing to be panned across the screen in a 512 dot horizontal x 483 line vertical array. Presuming that the horizontal origin is 0, the leftmost portion of the B size drawing from *Stack I* will appear on the screen. If the pan button is depressed and the *joystick* is moved to the left, the image will move to the left, and as the raster process sweeps to the right, the image from *Stack II* will enter the screen at the right. When 512 pixels on a scan line are presented on the screen, the sweep process is completed for a total of 483 lines vertically. Continuing to move the joystick to the left will cause the image to move to the left until, at the left-most position of the joystick, only the image from memory *Stack II* will appear on the screen.

To have a *continually panning image* with horizontal wrap around, the RATE key is depressed and the joystick is moved horizontally off-center, either to the left or the right, and the image will move to the left or the right at a speed across the screen which is proportional to the deflection of the joystick from its center position.

Likewise, *vertical panning* can be achieved by moving the joystick up or down. Since there are effectively 512 horizontal lines in the memory plane, but only 483 lines can be viewed on the screen, there are 29 lines in memory which cannot be viewed on the screen for a given video frame. Panning will expose these remaining 29 lines at the bottom of the screen. Continuing to pan vertically will expose more than these bottom most 29 lines. The additional lines will come from the top of the image in memory due to vertical wrap around.

There are several other possible memory configurations in SUPEROAM, including a 1024H x 1024V x 2 bit deep array. The latter is selectable automatically with the command SUP1 to invoke it and SUP0 to return the memory array to 512 x 512 x 8.

1.3 ANTI-ALIASED VECTORS

The AED 767 has a unique *anti-aliasing* feature which *eliminates the visible steps* or "jaggies" common to raster graphic vectors. Anti-aliasing can be performed by a host computer and the resultant smooth vectors can be transmitted to a multi-color terminal pixel by pixel. However, excess host processing and increased data transmission are the price for such smooth vectors. With the AED 767 all anti-aliasing of vectors in *eight colors* (black, red, green, yellow, blue, magenta, cyan, and white) *against a black background* are automatically selectable and generated within the terminal. Either the anti-alias key can be depressed on the terminal or the AAV command may be issued by the host to invoke this feature. From then on, all vectors generated within the AED 767 will be anti-aliased with a 16 color ramp for each of the aforementioned 8 colored vectors.

For anti-aliasing colors other than the default 8 colors to a black screen background or for anti-aliasing any eight colored vectors to any single non-black screen background, the user is provided the facility to program the $8 \times 16 = 128$ anti-aliased vector colors as required. Invoking *anti-aliasing* requires a total of 8 memory planes in the AED 767 and the upper 128 colors out of 256 are devoted to anti-aliasing.

1.4 BLUE LINE GRID

Another unique feature in the AED 767 is a blue grid which does *not* involve the use of a separate memory plane. Grid spacing of 8, 32 and 64 pixels in both X and Y, with blue line intensities corresponding to the grid line spacing, is provided. To turn on the blue grid the user may depress the special key, or the program can call it with the BLG command. The blue grid overall intensity has a separate control to suit the needs of the user. Furthermore, the joystick cursor (which normally moves in single pixel increments) can be programmed (SCP command) to move in jumps between intensified lines at spacings of 8, 32, and 64 pixels. This feature is valuable in CAD/CAM applications.

2. SPECIFICATIONS

2.1 TERMINAL SPECIFICATIONS (2-1)

- 2.1.1 Physical (2-1)
- 2.1.2 Electrical Power (2-1)
- 2.1.3 Environmental (2-2)
- 2.1.4 Mechanical (2-2)
- 2.1.5 “Input Only” Devices and Ports (2-2)
 - Keyboard
 - Joystick
- 2.1.6 Input/Output Ports (2-4)
 - Serial Ports
 - Parallel I/O Port
- 2.1.7 Video Output (2-7)
- 2.1.8 Display Processor (2-8)
 - Microprocessor
 - Video Memory
 - Color Look-Up Table
 - Firmware Reference
 - Execution Speed of Commands
 - Microcode

2.2 MONITOR SPECIFICATIONS (2-10)

- 2.2.1 Features (2-10)
- 2.2.2 Electrical (2-11)
 - Electrical Power
 - Video Input
 - Video Amplification And Display
- 2.2.3 Mechanical (2-12)

2.3 DEC (Digital Equipment Corp.) COMPATIBLE INTERFACES (2-13)

- 2.3.1 Physical (2-13)
- 2.3.2 Electrical (2-13)
- 2.3.3 Input (2-14)

2 SPECIFICATIONS

This chapter contains performance specifications for the terminal (base unit, display driver), the monitor (display unit) and lastly - the UNIBUS and Q-BUS interfaces to DEC PDP-11 and LSI-11 computers, respectively.

2.1 TERMINAL SPECIFICATIONS

2.1.1 Physical

	NOMINAL			
	WEIGHT	HEIGHT	WIDTH	DEPTH
AED 512/767 WITH 13" COLOR MONITOR:	60 lbs 27.2 kg	16" 40.6 cm	22.8" 57.9 cm	30" 76.2 cm
AED 512/767 WITHOUT COLOR MONITOR:	25 lbs 11.3 kg	3.5" 8.9 cm	22.8" 57.9 cm	30" 76.2 cm
AED 512/767 RACK MOUNT WITHOUT COLOR MONITOR:	25 lbs 11.3 kg	3.5" 8.9 cm	19" 48.3 cm	26.5" 67.3 cm
AED 512/767 REMOTE KEYBOARD:	10 lbs 4.5 kg	3.0" 7.6 cm	20.1" 51.1 cm	9.8" 24.9 cm

2.1.2 Electrical Power

MODEL	A.C. VOLTS	AMPS	FREQUENCY	ALL TOLERANCES
512A:	115	2	50/60 Hz	+/- 10%
512B:	230	1	50/60 Hz	+/- 10%
767A:	115	2	50/60 Hz	+/- 10%
767B:	230	1	50/60 Hz	+/- 10%
NOMINAL POWER SUPPLY OUTPUT VOLTAGES:		AED 512 +5, -5, +12 -12 VDC	AED 767 +5, -5, +12, -12, -2 VDC	

2.1.3 Environmental

	OPERATING	STORAGE
HOT/COLD TEMP CYCLING:	45 C/-5 C	60 C/-30 C
SHOCK:	2.0 G, 20 ms	3.0 G, 20 ms
ALTITUDE:	14,000 ft.	40,000 ft.
VIBRATION:	1.0 G (5-200 Hz)	2.0 G (5-200 Hz)
RELATIVE HUMIDITY:	50 C, 80% FOR 24 HOURS	

2.1.4. Mechanical

	AED 512	AED 767
BASE COLORS:	CARBIDE BLACK BEIGE #27786	DARK BROWN NO. 20040 BEIGE #27786
KEYBOARD COLORS:	CARBIDE BLACK BEIGE #27786	DARK BROWN #20040 BEIGE #27786
FACE PLATE:	PLEXI-GLASS 'G' ACRYLIC SHEET .125" THICK, GREY #2064	
HOUSINGS:	ALUMINUM CRT COVER 5052-H32 (.125" THICK) ALUMINUM KEYBOARD, BASE; ALL 5052-H32 (.090" THICK)	
SLIDES:	(RETMA Standards)	
RACK MOUNT:	19" (48.3 cm) STANDARD RACK MOUNT	
CLOSED SIZE:	23" (58.4 cm) DEPTH (FRONT TO REAR)	
EXTENSION RANGE:	44" (1.12 meters) DEPTH (FRONT TO REAR)	
CONFIGURATION:	REFER TO OUTLINE DRAWINGS, FIGURE 2.1.	

2.1.5 "Input Only" Devices and Ports

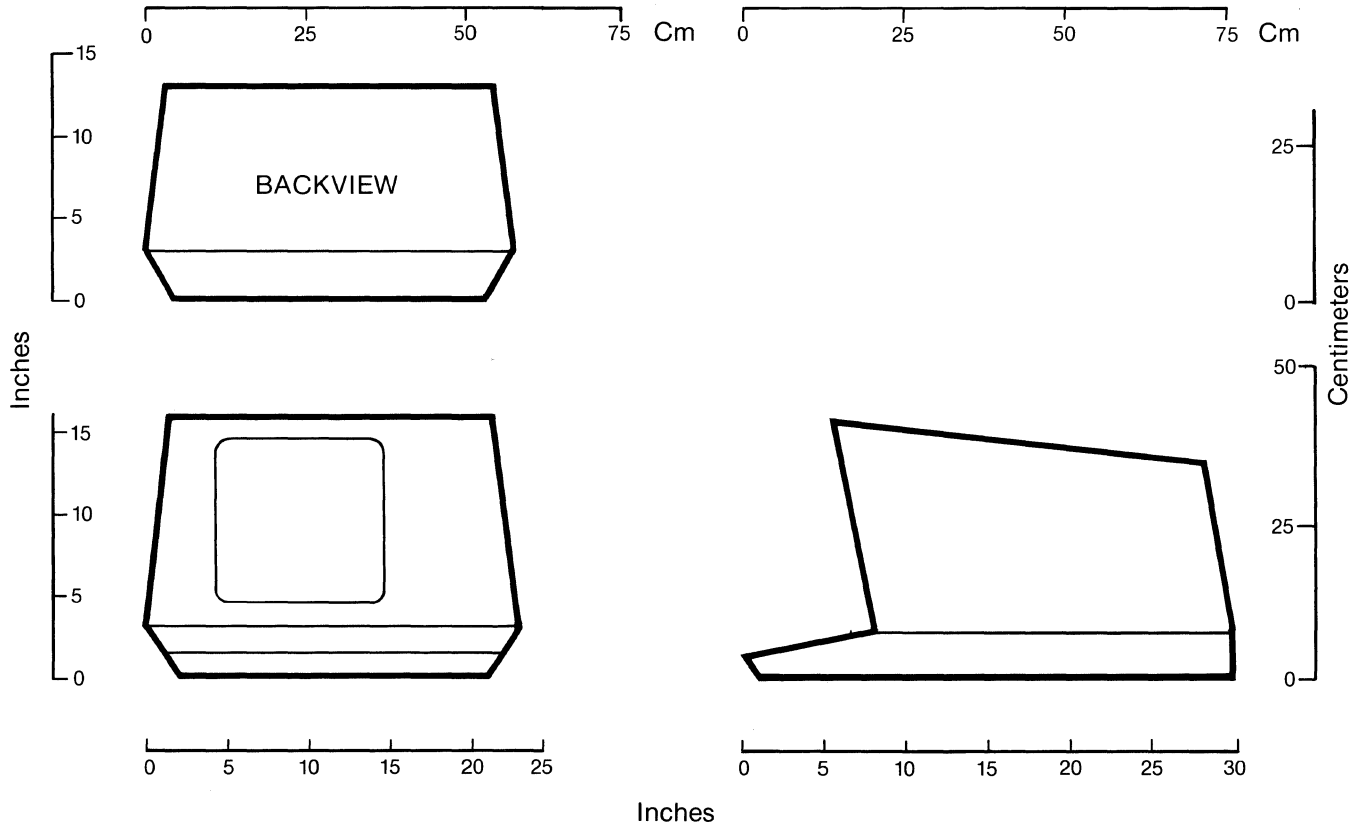
► **KEYBOARD**

NO. KEYS: 86, n KEY ROLLOVER

NO. ASCII CHARACTERS: 127

LENGTH OF REMOTE KEYBOARD CABLE: 15'(4.6 m) MAX

AED 512/767 Tabletop with 13" Monitor



AED 512/767 Rackmount with 19" Monitor

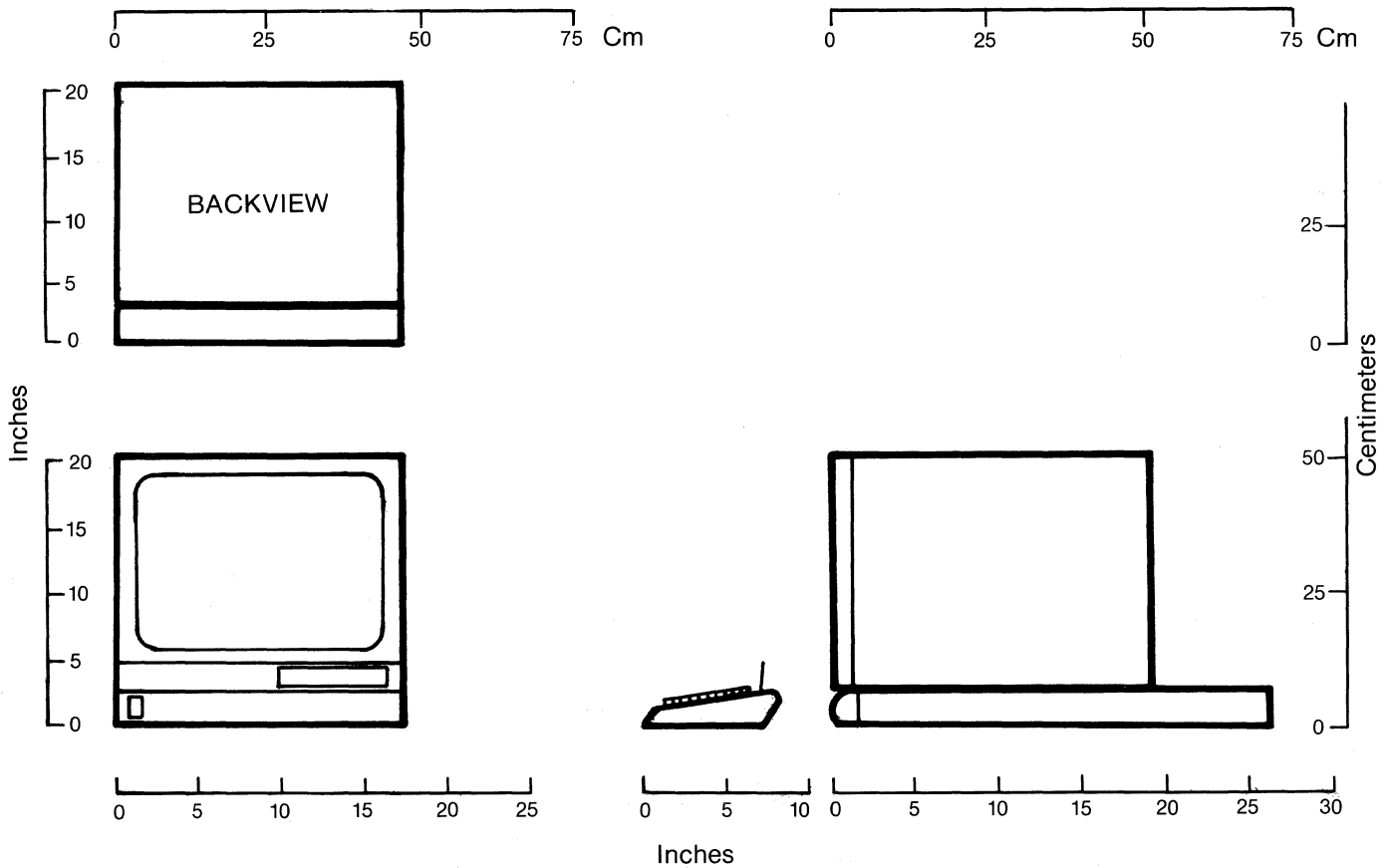


Figure 2.1

AED 512/767 Outline Dimensions

► **JOYSTICK**

**RANGE OF ANALOG
TO DIGITAL
CONVERTER**

OUTPUT: 000-3FF (HEX). VARIABLE IN X,Y

MODES: CURSOR CONTROL, PAN, SCROLL, RATE.

2.1.6 Input/Output Ports

► **SERIAL PORTS**

PORTS: 2 ; 1 HOST, 1 AUXILIARY

HOST: RS232-C or 20 MILLIAMP CURRENT LOOP

AUXILIARY: RS232-C ONLY

CONNECTORS: RS232-C 25 PIN "D" TYPE CONNECTOR, FEMALE ON TERMINAL

BAUD RATE: 300-19.2K PLUS EXTERNAL CLOCK, EACH PORT INDEPENDENTLY SETTABLE THROUGH COMM SWITCH OR SOFTWARE. EXTERNAL CLOCK PERMITS HIGHER OR LOWER RATE.

BUFFER SIZE: 256 BYTE STACK LOCATED IN THE MICRO-PROCESSOR RAM, CONTROLLED BY THE 6502A.

MODES: FULL DUPLEX/HALF DUPLEX, SETTABLE THROUGH COMM SWITCH OR SOFTWARE.

CURRENT LOOP: 20 mA CURRENT LOOP ON THE HOST PORT ONLY. HOST OR TERMINAL CAN SUPPLY THE CURRENT.

► **PARALLEL I/O PORT**

PORTS: 1 DMA PORT

DATA TRANSFER RATE: 250K BYTES OR WORDS/SEC TYPICAL, MAX SPEED = 3M BYTES/SEC

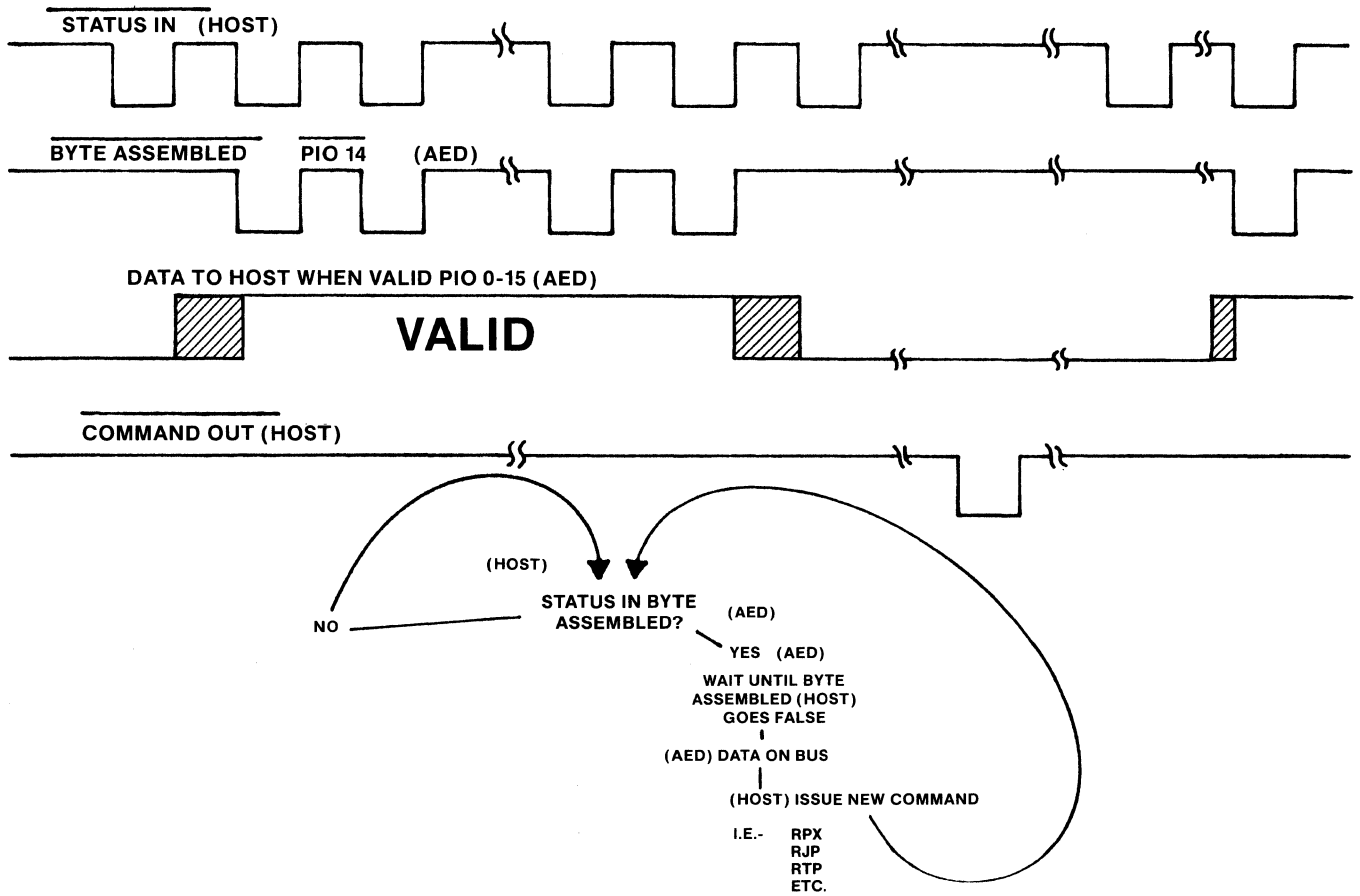
DMA CABLE LENGTH: 6'-50' (1.8 - 15.2 m)

MODES: DMA (DIRECT MEMORY ACCESS) MODE - VERY HIGH SPEED MEMORY TRANSFER DIRECT FROM HOST TO VIDEO MEMORY, COMMUNICATION THROUGH PARALLEL PORT.

COMMAND DMA - PARALLEL BYTE TRANSFERS UNDER PROGRAM CONTROL, COMMUNICATION THROUGH PARALLEL PORT.

TIMING DIAGRAM: SEE FIGURE 2.2

PARALLEL BYTES TO HOST



PARALLEL BYTES TO AED

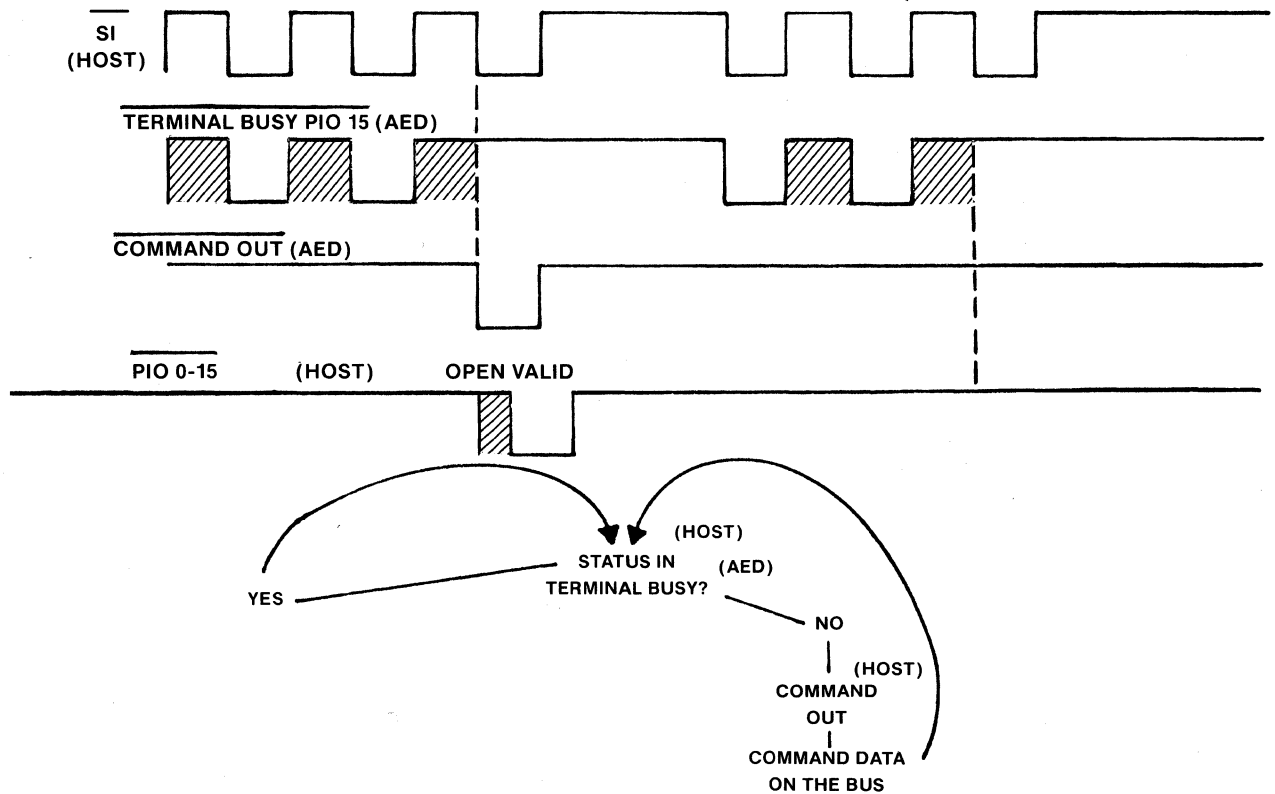
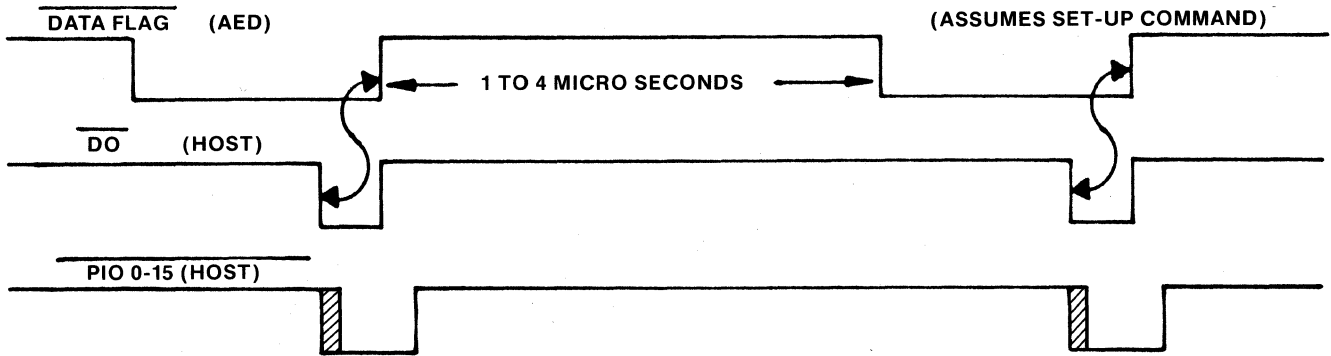


Figure 2.2.A
Timing Diagrams

DVMA OR DMA COMMANDS TO AED



DVMA READ FROM AED

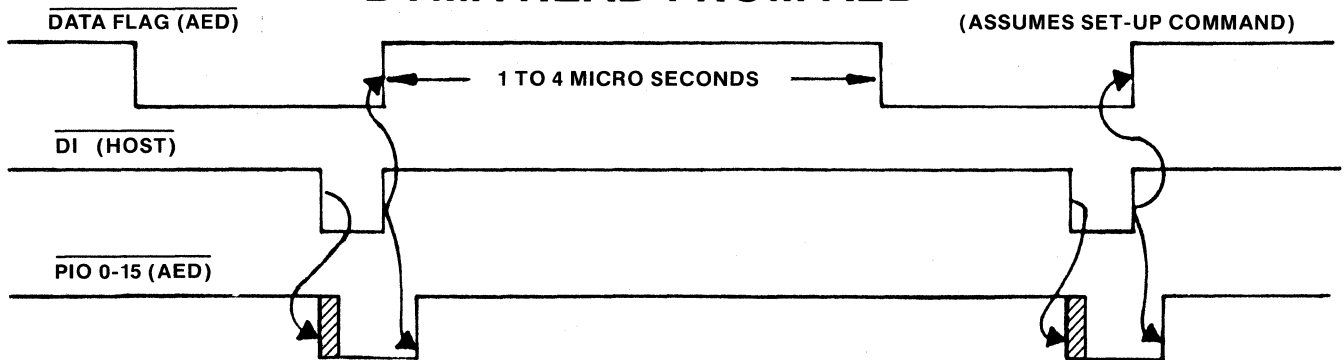


Figure 2.2.B
Timing Diagrams

TERMINATION: TYPICALLY 330/390 OHM PULLUP/PULLDOWN TERMINATORS WHICH ARE INSERTED AT BOTH ENDS OF HOST INTERFACE AND THE LAST TERMINAL, WHEN LOOP-THROUGH CONNECTED.

CONNECTOR: 50 PIN FLAT RIBBON CONNECTOR

HOST COMPUTERS: AED SUPPORTS DEC ONLY, FOR DMA INTERFACING. LSI-11/XX, PDP-11/04-70, VAX SERIES.

INTERFACES: Q-BUS (DUAL WIDTH), CARD PLUGS INTO PROPER DMA POSITION IN LSI-11 BACKPLANE. UNIBUS (QUAD WIDTH) CARD PLUGS INTO SPC SLOT OF PDP-11 BACKPLANE.

2.1.7 Video Output

MONITOR STANDARDS: PLEASE REFER TO SECTION 2.2, AED MONITOR SPECIFICATION.

REFRESH RATES AND VIEWING WINDOW:

AED 512		AED 787	
REFRESH RATE #	WINDOW, PIXELS x LINES	REFRESH RATE #	WINDOW, PIXELS x LINES
30 Hz	*512H x 483V (NTSC PROM)	40 Hz	768H x 483V 640H x 483V 512H x 483V
25 Hz	512H x 512V (CCIR PROM)	30 Hz	768H x 483V *640H x 483V 512H x 483V
		33.5 Hz	768H x 575V 640H x 575V 512H x 575V
		25 Hz	+768H x 575V 640H x 575V 512H x 575V

NOTES: Refresh Rate and Window are Switch selectable. See Chapter 3 Installation.

Frame referenced, nominal value
* NTSC Compatible
+ CCIR Compatible

INTERLACE: YES YES

2.1.8 Display Processor

► MICROPROCESSOR

TYPE:	6502A	
CLOCK RATE:	2 m Hz	
INSTRUCTION SPEED:	1 usec/Instruction	
	AED 512	AED 767
RAM SPACE:	5K BYTES	10K BYTES
ROM SPACE:	16K BYTES	20K BYTES
USER RAM (DOWN-LOADABLE):	2K BYTES	7K BYTES
RAM ADDRESS:	3328~10239(DECIMAL)	3072~5119 (DECIMAL)

► VIDEO MEMORY

SIZE:	4116 IC (32K/PLANE)	4164 IC (128K/PLANE)
SPEED:	250 ns	200 ns
REFRESH RATE: PIXELS	REFRESH EVERY 64 PIXELS	REFRESH EVERY 64 PIXELS
CONFIGURATION RANGE:	1 TO 8 MEMORY PLANES	4 OR 8 MEMORY PLANES
WRITE MASK BITS:	8	8
READ MASK BITS:	8	8

► COLOR LOOK-UP TABLE

SIMULTANEOUS DISPLAYABLE COLORS:	256	256
COLOR PALETTE:	16.8 MILLION	16.8 MILLION
NUMBER OF BITS INPUT:	8	8
NUMBER OF BITS OUTPUT:	24	24
NUMBER OF CHANNELS:	3	3
BITS PER CHANNEL:	8	8

**STARTING ADDRESS (HEX)
OF COLOR TABLE:**

	AED 512		AED 767	
	STARTING	ENDING	STARTING	ENDING
RED	1C00	1CFF	3C00	3CFF
GREEN	1D00	1DFF	3D00	3DFF
BLUE	1E00	1EFF	3E00	3EFF

767 READ BACK LOCATIONS

RED	800	8FF
GREEN	900	9FF
BLUE	A00	AFF

**CAN READ AND WRITE IN THE
SAME ADDRESS?**

YES

NO

In 767, register addresses 3C00 through 3EFFF are write only, but a copy of the color arguments can be read from RAM addresses 800 through AFF. Corresponding AED 512 registers can be read directly.

COLOR IMAGE IN RAM

STARTING ADDRESS: 800 (HEX) 800 (HEX)

NUMBER OF COMMANDS: 92 98

(PLUS TEKTRONIX 4010-4015
EMULATION COMMANDS)

► **FIRMWARE REFERENCE**

For a copy of the Firmware Version Reference contact AED Headquarters. Ask for the current "Firmware Log" listings.

► **EXECUTION SPEED OF COMMANDS**

AVERAGE WRITING SPEED: 7 usec/pixel

**AVERAGE VECTOR WRITING
SPEED:** 3.5 usec/pixel

**AVERAGE ANTI-ALIASED
VECTOR WRITING SPEED:** 13.5 usec/pixel

AVERAGE SET-UP TIME: 80 usec

► **MICROCODE**

To obtain a copy of the "Microcode Source Listings", you must sign a Non-Disclosure Agreement with AED. For more information please contact AED Headquarters in Sunnyvale.

2.2 Monitor Specifications

MITSUBISHI MODEL C-3419 AND C-3919

Mitsubishi Electric, MODEL C-3419 and 3919 Series Color Display Monitors used with AED 512/767 are high resolution color display modules, for clear display of 2000 characters, or up to 441,600 pixels of graphic output. These modules are equipped with an IN LINE GUN/SHADOW MASK CRT (cathode ray tube) and PCBs (printed circuit boards) with solid state active elements. The MODEL C-3419/3919 Series features stable convergence, easy maintenance, and compact style. The standard model accepts analog inputs for RGB and composite Sync signals. This model complies with U.S. Department of Health, Education and Welfare X-radiation Safety Rules, applicable at the time of manufacture. The most obvious difference between the two monitors is that the 3419 has a nominal 13 inch CRT while the 3919 has a 19 inch.

2.2.1 Features

COMPACT STYLE

The 3419 model is sufficiently compact that it can be used as a stand-alone monitor (AED 512R or AED 767R), or be built into a stylized cabinet (AED 512T/767T).

ALL SOLID STATE EXCEPT FOR CRT

All active elements, except for the CRT, are solid state elements e.g. IC or Transistor.

EASY MAINTENANCE

PCBs can be replaced without use of tools and most parts can be checked and replaced without disassembling any construction.

ANTI-SPARK CIRCUIT

All circuits are designed to prevent damage caused by spark in the CRT.

STABLE CONVERGENCE

Self-convergence assemblies are mounted on the CRT. Complicated convergence procedures are not necessary, because electrical convergence circuits are not used.

2.2.2 Electrical

► ELECTRICAL POWER

	A.C. VOLTS	VOLTAGE TOLERANCE	VOLT-AMPERES	FREQUENCY
C-3419 (13''):	100-120 OR 200-240, TAP SELECTABLE	+/-10%	180	50-400 Hz
C-3919 (19''):	100-120 OR 220-240, TAP SELECTABLE	+/-10%	200	50 or 60 Hz

► VIDEO INPUT

TERMINATION: 75 Ohms or High Impedance are selected by termination switches.

CONNECTORS: BNC connectors for all inputs

TYPES OF INPUTS:
Red - Video Signal
Green - Video Signal or Composite
Blue - Video Signal

These three signals are positive white. Sync shall be composited with the green video. Separate sync input shall be applied with the green video at Sync Input in case the video signal is without sync.

INPUT LEVELS: 0.7-1.5 Vp-p for R,G,B inputs
1.0-5.0 Vp-p for Sync signal

TIMING REQUIREMENTS: See Figure 2.3

SCANNING FREQUENCY: Vertical Frequency 40-90 Hz
Horizontal Frequency 15.5-24 kHz
(TAP CHANGEABLE)

CATHODE RAY TUBE (CRT): Self-convergence type, dot-phosphor shadow mask tube and in-line electron gun. Phosphors are Red, Green and Blue. For low refresh rates, long persistence phosphors are recommended to reduce flicker.

AMBIENT TEMPERATURE: Ambient Temperature during equipment operation should be -5 to +40 C.

WARM-UP TIME: Warm-up time is 20 minutes max. At the end of the warm-up period, no adjustments are necessary to meet the specifications contained herein.

PACKAGE ENVIRONMENT: This equipment withstands room air temperature of -30 C to +60 C and 20 in. (50 cm) free drops encountered during transportation, handling and storage. It also withstands Relative Humidity of 0% to 95%.

BLANKING TIME: Horizontal - LESS THAN 9.5 usec Vertical - LESS THAN 0.8 usec

VIDEO AMPLIFIER: The video amplifier is a linear amplifier which drives the cathode of the CRT. Video signals are compatible with the timing requirements of EIA STANDARD RS-170.

The peak-to-peak input signal amplitude will be between 0.7 Volts and 1.5 Volts.

Composite video signal shall be composed of approximately 70% video and 30% sync amplitude.

- A. Frequency Response : +/-3 db or better between 50 Hz to 20 mHz
- B. Pulse Response : Rise and Fall Times are shorter than 20 ns
- C. Differential Gain : Better than 5%
- D. Black Level : Pedestal clamp Stability circuits are provided. BLACK level is maintained within 1% at any Average Picture Level of 10% to 90%.

CONVERGENCE: Less than 0.5 mm in a centrally located area bounded by a circle whose diameter is equal to picture height. Elsewhere the deviation is than 0.8 mm.

RASTER SIZE REGULATION: Raster Size change caused by change of CRT beam current between 0-200 uA is less than 4 mm for C-3419 and C-3919.

LINEARITY AND GEOMETRY: Linearity measured and calculated by the following formula is less than +/- 7%.

Formula: $((MAX-MIN)/MEAN) \times (1/2) \times 100\%$

Raster distortion is less than 2% of raster height.

MAXIMUM EFFECTIVE SCREEN SIZE:

	C-3419/24	C-3919
HORIZONTAL	10.6 in (270 mm)	14.9 in (380 mm)
VERTICAL	7.7 in (195 mm)	11.0 in (280 mm)
DIAGONAL	13.1 in (333 mm)	19.0 in (483 mm)

2.2.3 Mechanical

PANEL CONTROLS: THE FOLLOWING REAR PANEL CONTROLS ARE PROVIDED —

BRIGHTNESS CONTROL
GAIN CONTROL
DEGAUSS SWITCH
POWER SWITCH

- CONFIGURATION:** REFER TO OUTLINE DRAWINGS, FIGURE 2.1.
- IDENTIFICATION AND MARKING:** THE FOLLOWING MARKINGS ARE PROVIDED —
1. H.E.W. WARNING LABELS ON THE CRT AND CHASSIS.
 2. HIGH VOLTAGE WARNING LABELS ON THE CHASSIS OR THE CABINET.
 3. RATING LABEL ON THE CABINET OR CHASSIS TO SHOW POWER SOURCE, MODEL NUMBER, ETC.
 4. SERIAL NUMBER LABEL ON THE COVER OR CHASSIS.

2.3 DEC (Digital Equipment Corporation) COMPATIBLE INTERFACES

The AED 512 and AED 767 Terminals may be interfaced to a Local Host Computer through a Direct Memory Access (DMA) Interface. This connection provides a high speed channel to move data between the host and terminal memories. The DMA interface also provides a rapid means of transferring terminal status to the host and host commands to the terminal. The DMA cards may also be used for programmed I/O transfers.

AED currently provides and supports DMA interfaces for the Q-bus and Unibus (registered trademarks of DIGITAL EQUIPMENT CORPORATION). These interfaces mount inside the host computers, draw DC power from the host Power Supplies and are connected to the AED 512/767 by a 50 pin/wire flat ribbon cable. The Q-bus interface is contained on a dual-width card. The Unibus interface is contained on a quadwidth card. Vector Interrupts and Device Addresses are selectable by the user.

2.3.1 Physical

	LSI-11 Q-BUS	PDP-11 UNIBUS
LENGTH:	8 3/8 in./(211 mm)	10 3/8 in./(260 mm)
WIDTH:	5 1/14 in./(130 mm)	8 7/16 in./(211 mm)
HEIGHT:	1/16 in./(1.0 mm)	1/16 in./(1.0 mm)
WEIGHT:	6.5 oz./(184 gm)	11 oz./(311 gm)

2.3.2 Electrical

HOST POWER REQUIRED:	+5 V@ 2.5 AMPS	+5 V@ 2.8 AMPS
HOST UNIT LOADS:	1 DC/2.5AC	1 DC/4.5 AC
INTERFACE CABLE LENGTH:	6'/(1.8 m) STANDARD	6'/(1.8 m) STANDARD
	50'/(16 m) MAXIMUM	50'/(16 m) MAXIMUM

2.3.3 Input (Address In Octal)

	LSI-11 Q-BUS	PDP-11 UNIBUS
DATA/(ADDRESS) LINES:	8 or 16/(18)	8 or 16/(18)
CONTROL & STATUS LINES:	16	16
STANDARD DEVICE CSR ADDRESS:	164040	164040
RANGE OF DEVICE CSR ADDRESSES:	164000-177770	164000-177760
INTERRUPT/EXTENDED ADDRESS:	CSR+2	CSR+2
BUS ADDRESS:	CSR+4	CSR+4
WORD COUNT ADDRESS:	CSR+6	CSR+6
STANDARD INTERRUPT VECTOR ADDRESS:	150	150
RANGE OF INTERRUPT VECTOR:	2-374	2-374

3. INSTALLATION

3.1 UNPACKING (3-1)

3.2 INSPECTION (3-2)

3.3 ELECTRICAL POWER-UP (3-3)

3.4 VERIFICATION OF OPERATION (3-8)

3.4.1 Basic Verification (3-8)

3.4.2 Additional Operations (3-9)

3.5 COMMUNICATIONS (3-12)

3.6 SOFTWARE (3-12)

3.7 OVERNIGHT OR CONTINUOUS USE (3-12)

3.8 SWITCH SETTINGS (3-12)

3.9 GENERAL INSTALLATION CHECK LIST FOR AED 512/767 AND HOST CPU (3-17)

3.9.1 Troubleshooting (3-17)

3 INSTALLATION

3.1 UNPACKING

The AED 512/767 terminal, monitor, and DMA interface will be received, foam protected, in separate cartons. After carefully removing the contents (hardware and documentation), store the cartons and packing material in a safe place for future use should the equipment need to be shipped to a new site or returned to the seller for repair.

3.2 INSPECTION

Carefully inspect each carton of the shipment, comparing it with the shipping document and noting any physical defects on the Inspection and Installation Report at the front of this manual. Be sure to *advise your business office* of any damages and have them contact the seller. Return one copy of the inspection report to the seller. Under normal circumstances, there will be no dents, blemishes or scratches on the monitor or terminal surfaces.

Tip the monitor carefully from side to side, testing for loose hardware. Repeat the process for the terminal. If something sounds loose in either section, remove the screws from the cover(s) and inspect for internal damage. If there is no serious damage, replace and tighten loose nuts or screws and secure the cover. If serious damage has occurred DO NOT POWER UP THE UNIT, but contact the seller for advice.

Assuming the units are not damaged, *place the monitor atop the terminal* as shown in *Figure 3.1*. Align the rear edge of the monitor with the same of the terminal. *Connect the three coaxial cables* as shown in *Figure 3.1* and *3.1A*. Be sure that one of the cables provided connects RED (R) on the monitor to RED (R) on the terminal; likewise for G and B. If a single monitor is being used, the monitor termination switches should be in the 75 Ohm position. But if several monitors are being cascaded, connections should be made and termination switches set as in *Figure 3.2*.

3.3 ELECTRICAL POWER-UP

Next, inspect the utility cord coming from the terminal and the monitor to be sure it has the proper connector for your electrical service (AC utility power). Also, *look at the AED serial number labels* on the rear of the monitor and terminal to be sure the voltage you have available is that shown on the labels. If they are not the same, CALL THE SELLER before connecting power.

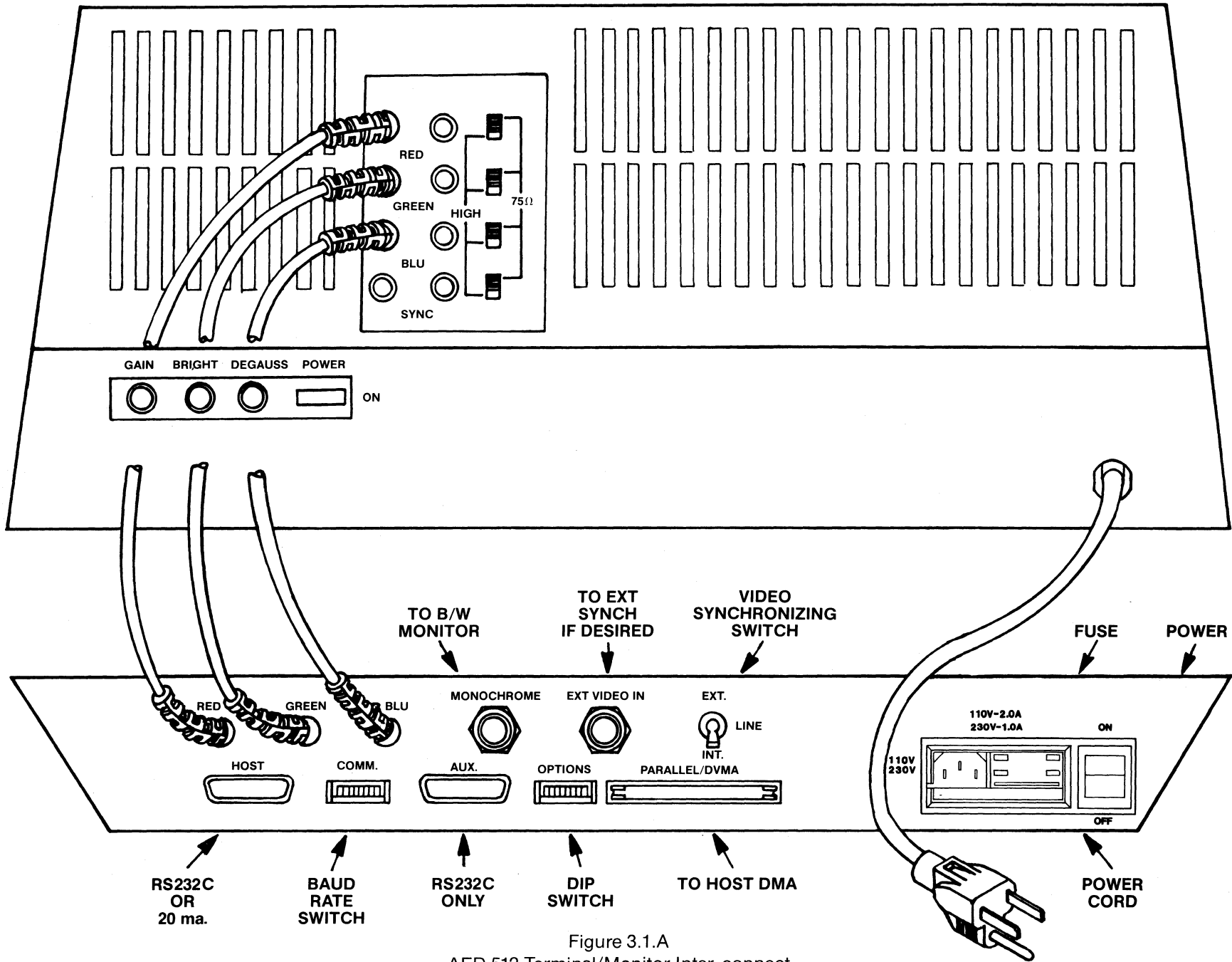


Figure 3.1.A
AED 512 Terminal/Monitor Inter-connect

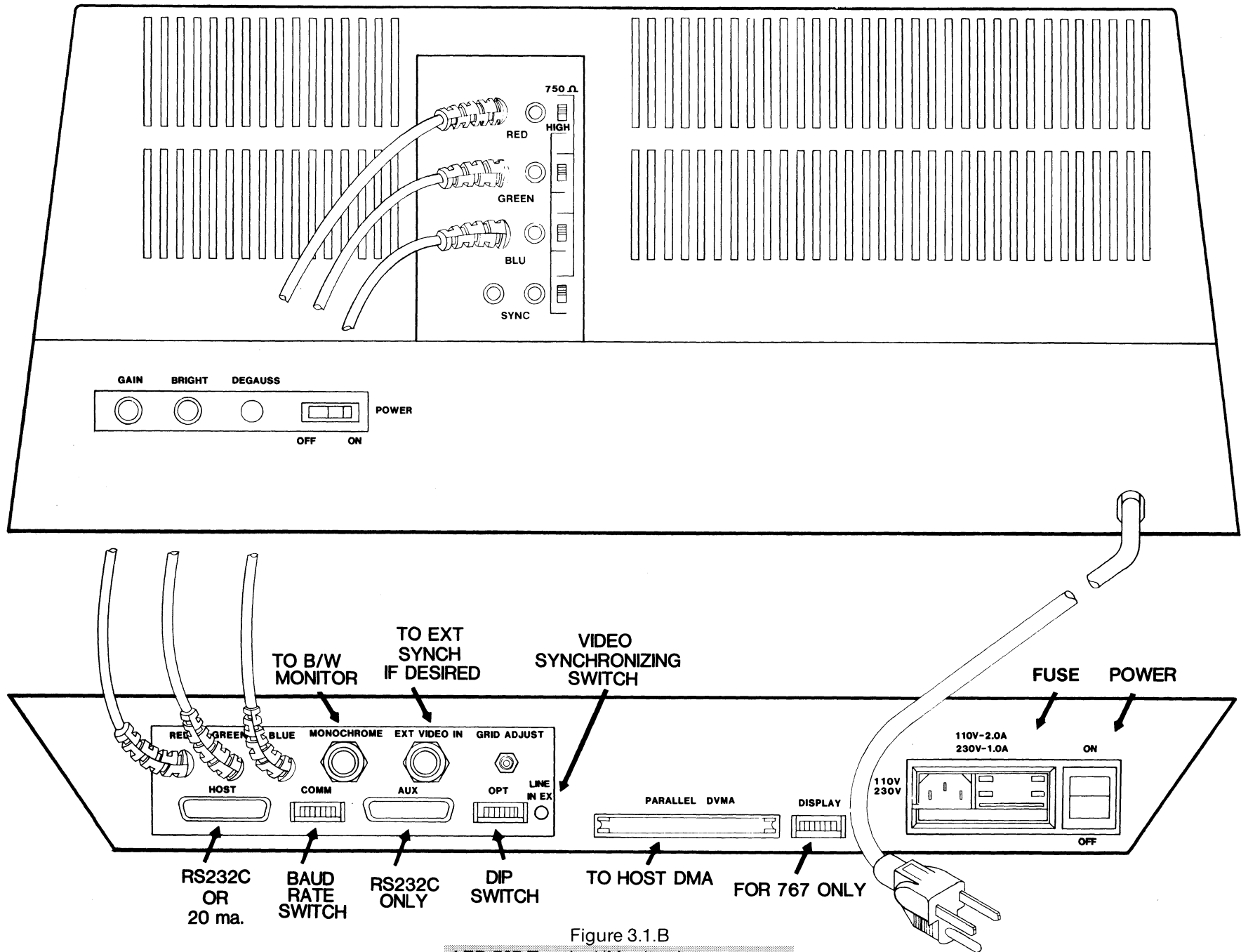


Figure 3.1.B
AED 767 Terminal/Monitor Inter-connect

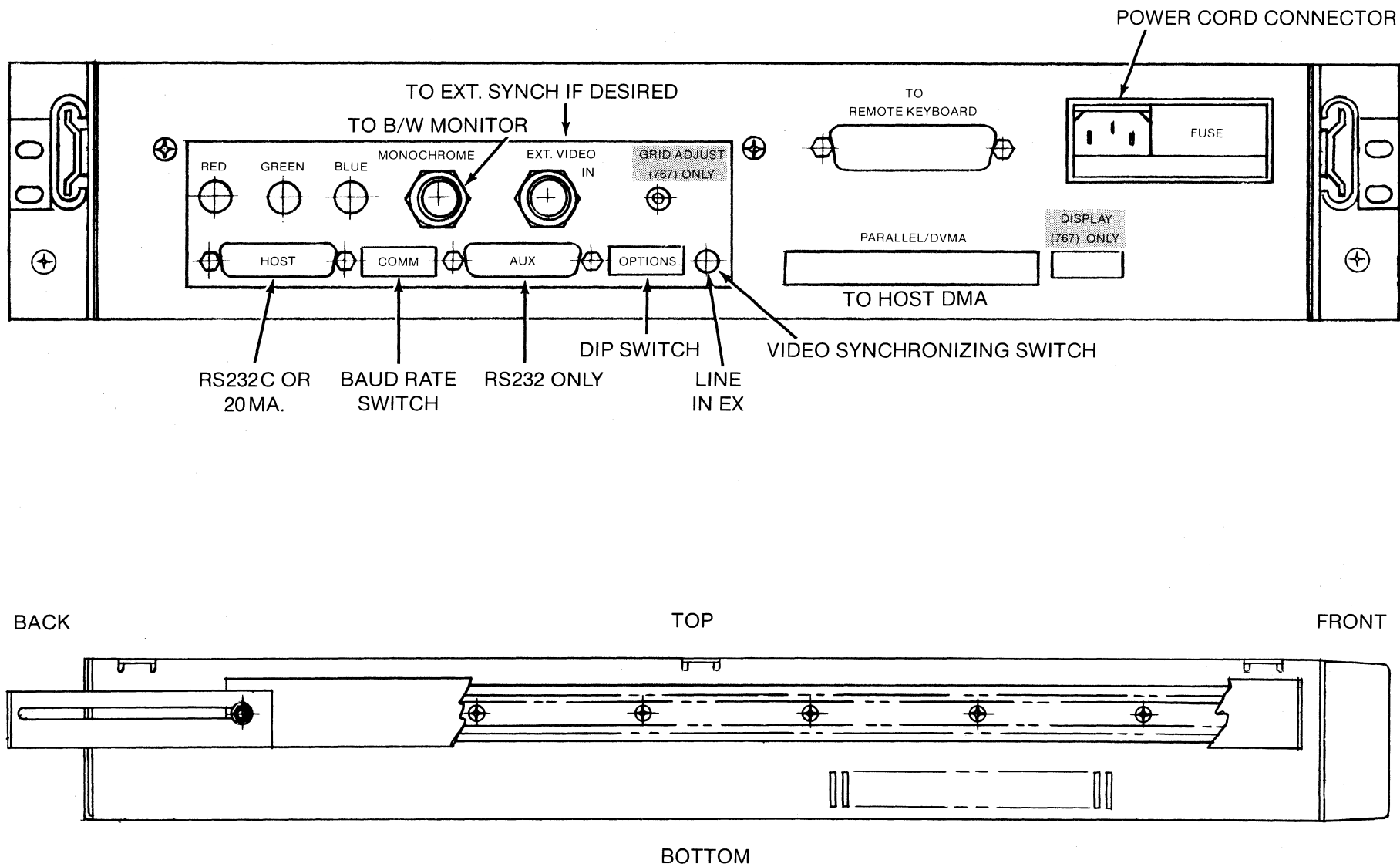
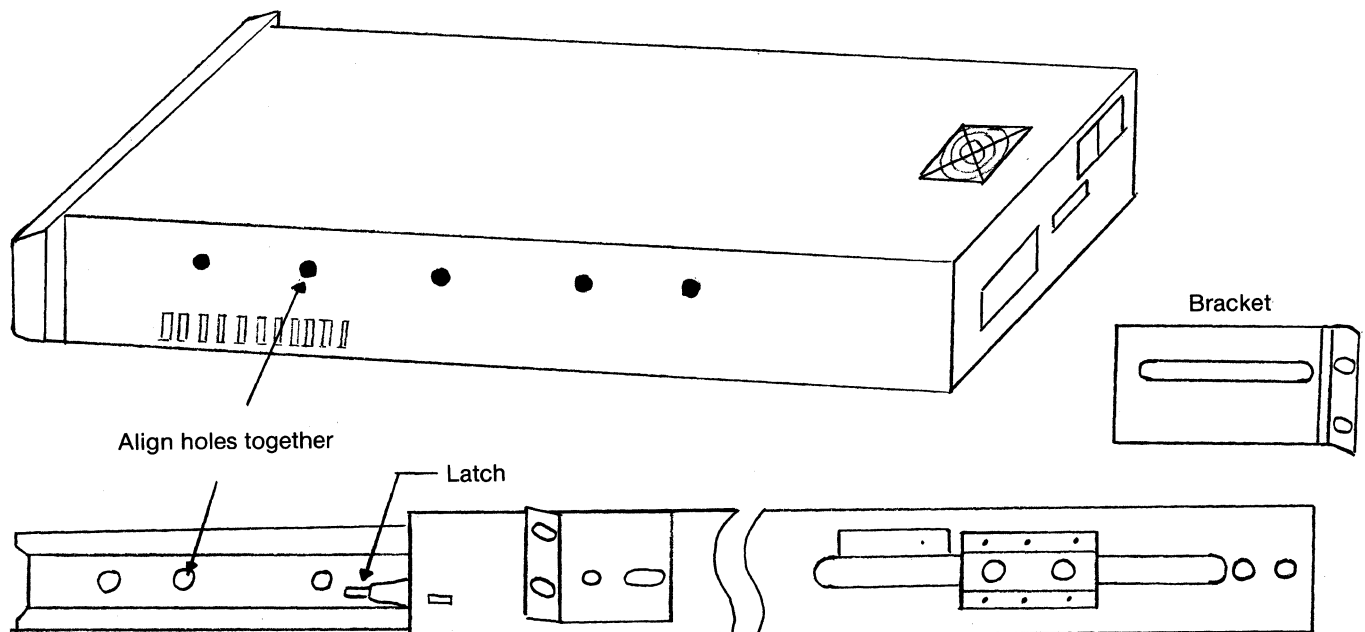


Figure 3.1.C
AED 512/767 Rack Mount

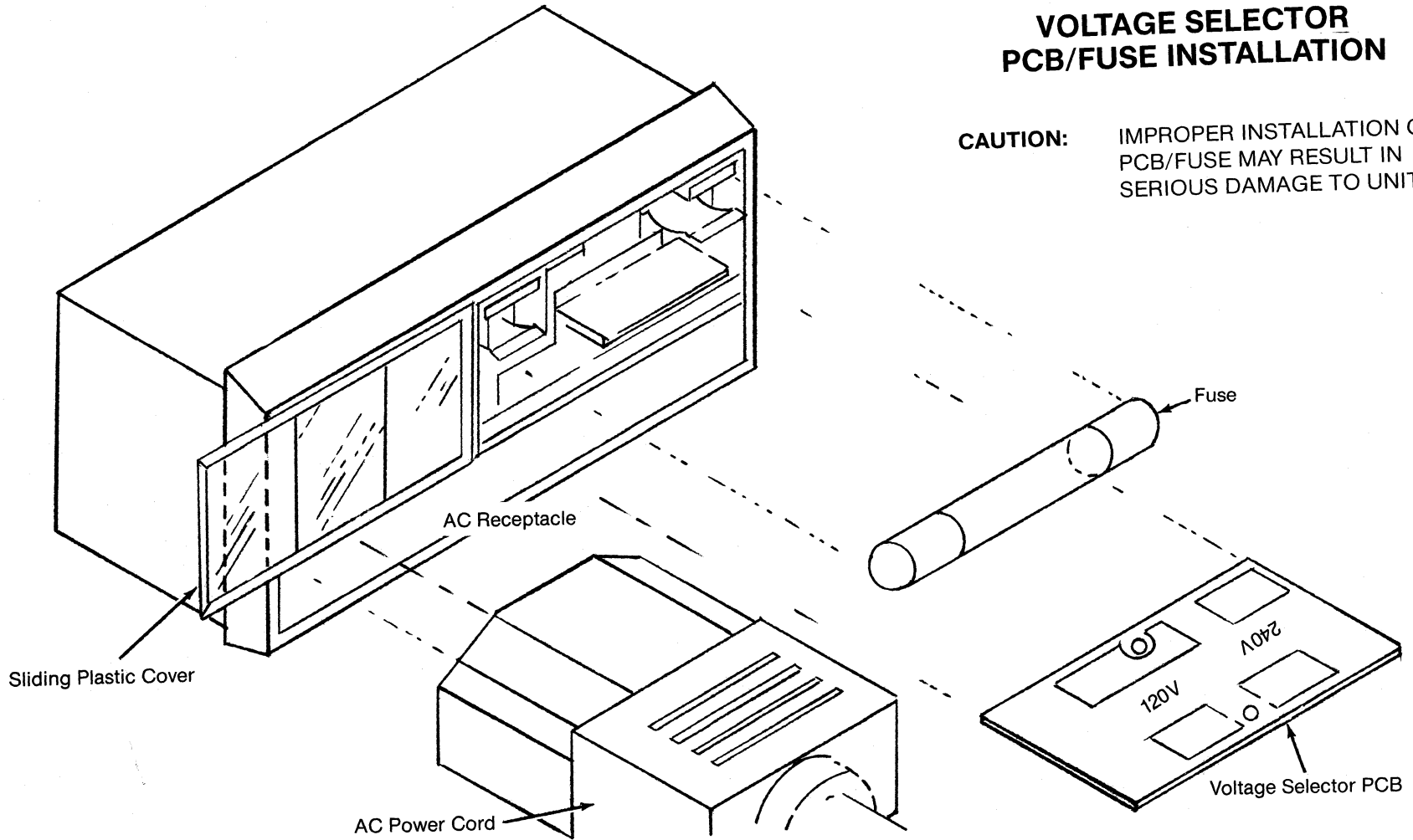
INSTRUCTIONS for the INSTALLATION of SLIDER ASSEMBLIES on AED 512/767 RACK MOUNT UNIT

- 1) Extend the two inner sliders completely and then remove the innermost slider by pushing the latch upward.
- 2) The latch should be pointing downward and toward the front of the unit.
- 3) Align the second hole of the inner slider with the second hole of the unit and secure the slider with four #10 screws.
- 4) Repeat the above three steps to the opposite side of the unit.
- 5) Install the bracket onto the outside rear of the outer slider, using a flathead screw in the outermost hole and a #8 screw in the other hole. Be sure screw heads are inside the slider. Attach washer, split washer and nut (in that order) to secure the screw to slider. (NOTE: The flange of the bracket should point out toward the nearest cabinet side.) **Do not tighten screws yet.**
- 6) Insert the outer slider into the cabinet and adjust the loose bracket to ensure a proper fit. Tighten the screws of the bracket.
- 7) Attach the outer slider to the inside of the cabinet with the bracket flanges.
- 8) Repeat steps 5, 6 and 7 to install the outer slider to the opposite side of cabinet.
- 9) Carefully slide the Rack Mount unit into the track of the middle slider (push latch up to release the slider) and make sure unit is completely within cabinet.



VOLTAGE SELECTOR PCB/FUSE INSTALLATION

CAUTION: IMPROPER INSTALLATION OF PCB/FUSE MAY RESULT IN SERIOUS DAMAGE TO UNIT.



For proper voltage configuration, insert voltage selector PCB into AC receptacle such that right reading viewed voltage is the desired voltage.

NOTE:
For 100V-120V operation, install 2 AMP fuse.
For 220V-240V operation, install 1 AMP fuse.

Figure 3.1 E

REAR PANELS OF MONITORS

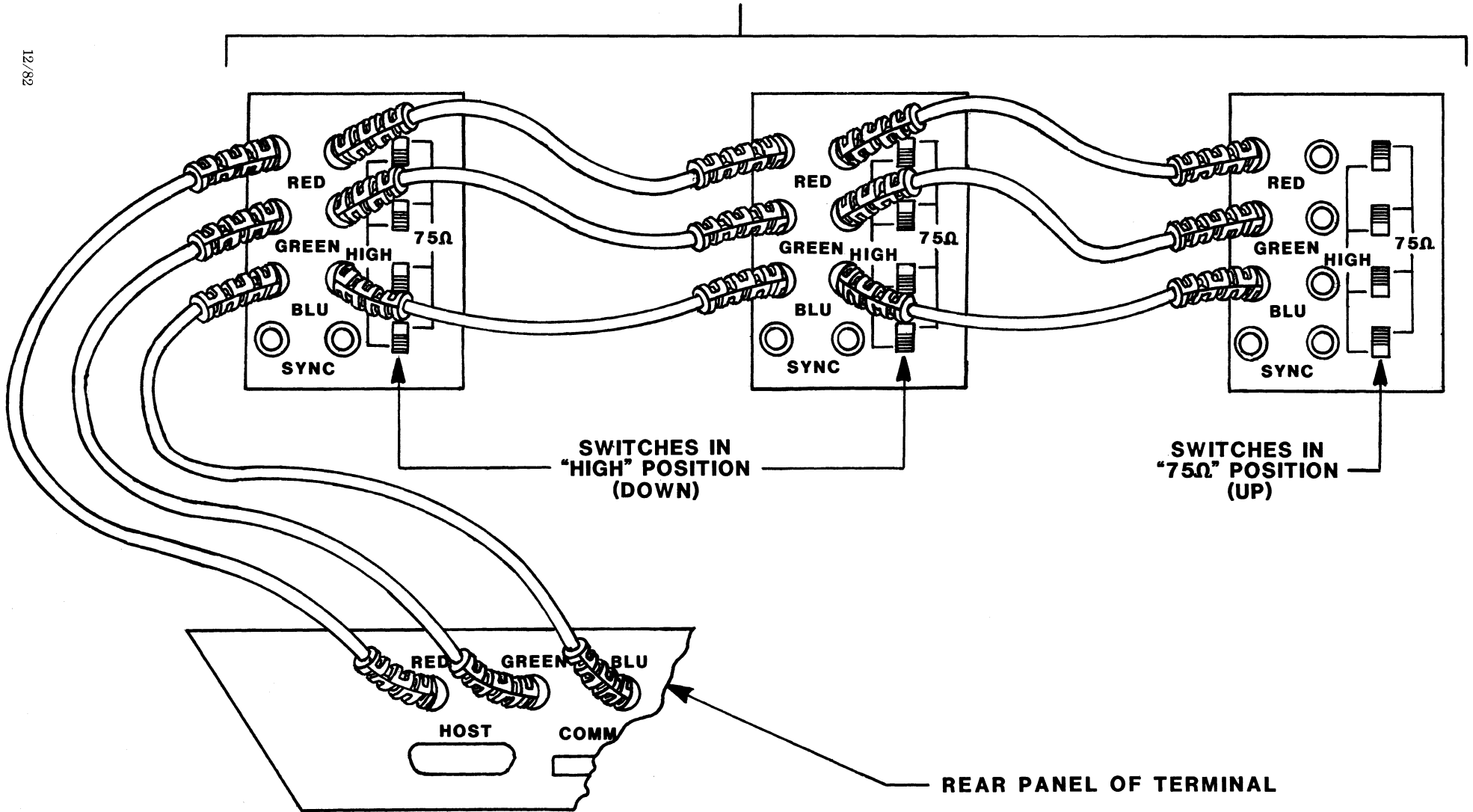


Figure 3.2
Connection of Daisy Chained Monitors

With the correct power applied to the AED 512/767 terminal and monitor, push the power switch on the left, rear surface of the terminal (per Figure 3.1) to ON (UP position) and also switch ON the monitor power (switch is located in rear). Within a second or two, an audible chirp sound will be heard indicating the terminal self-test diagnostics have been successfully completed.

If no chirp is heard, check the power switch and toggle it to the UP position. If the chirp is still not heard, recheck the AC power connections and try again.

Assuming the terminal chirps, a *terminal signature* message should be visible on the upper left hand corner of the monitor screen. The signature will be of the form —

AED (Model (Circuit Board V (Firmware (Horizontal x Vertical
Number) Revision Letter), Revision Number) displayable pixels, 767 only)

Examples are AED 512B V82.7 and AED 767A V82.7 768 x 575. The Firmware Revision Number will have the form YEAR. MONTH; these examples show a version released in 1982, the 7th month. If the signature message is not immediately visible, recheck the cables and make sure the monitor power is ON; depress twice the RESET key on the left top row of the keyboard and now the chirp and signature should be evident.

3.4 VERIFICATION OF OPERATION

Assuming the terminal chirps and the terminal signature appears, a *few quick tests should then be run from the keyboard*. The following description assumes a terminal having three or more memory planes. A machine with just one plane would have available only the colors black and red, using the standard (default) color table.

3.4.1 Basic Verification

First, depress the LOCAL key (to the right of RESET), followed by the CURSOR and ESC keys. A green dotted "X" cursor should appear on the monitor. Since depressing the ESC key, the terminal has been in the Interpreter Mode (767 must have CAPS LOCK asserted). Manually moving the joystick (at the right side of the control area) will cause the cursor to move. Place the cursor at a desired position and depress O (for Origin of circle; cursor will disappear) followed by A; a **red circle** should appear with the cursor at its center. Depressing I (Interior Fill) will fill the circle red.

Move the cursor to a new location, then perform the keystroke sequence:

```
C
CTRL B
O
SPACE
```

Note, CTRL B is achieved by *simultaneously depressing*, once only, the CTRL and B keys. A **green circle** of diameter equal to one-half that of the red circle will appear. Depressing I will fill the circle green but not go beyond any boundary surrounding the cursor.

Move the cursor to a third location, this one just outside the green circle. Do keystrokes:

C
CTRL D
O
RUB

A **blue circle** of diameter twice that of the red circle will appear. Depressing I will fill the circle blue, but not beyond any boundary surrounding the cursor.

To move the display about the monitor screen, depress PAN and move the joystick. Depress PAN again to disable the movement control.

Clear the screen and return the terminal to its turn-on condition (Alphanumeric Mode) by depressing RESET twice. This completes the few quick tests to verify correct installation and operation.

3.4.2 Additional Operations

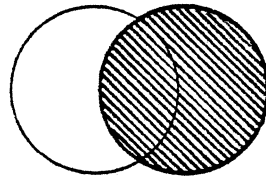
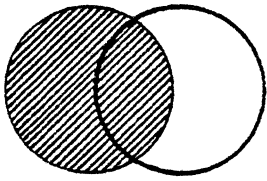
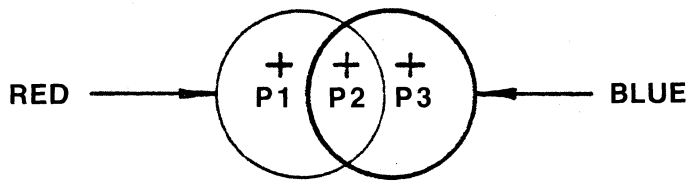
Depending upon how many memory planes are installed in the terminal, the circles can be drawn with various colors other than red by changing the color before depressing O. Color change is achieved, in the *Interpreter (Graphics) Mode*, by: noting the cursor is on the screen (if extinguished, depress CURSOR key); striking C (Color change); and then simultaneously depressing several keys to obtain one of the colors listed below (e.g., CTRL and D keys to get a blue circle).

NOTE: 767 must have CAPS LOCK key ON.

CIRCLE COLOR	KEYS DEPRESSED	PIXEL VALUE (COLOR TABLE ADDRESS)
BLACK	CTRL SHIFT P	0
RED	CTRL A	1
GREEN	CTRL B	2
YELLOW	CTRL C	3
BLUE	CTRL D	4
MAGENTA	CTRL E	5
CYAN	CTRL F	6
WHITE	CTRL G	7

To fill the complete background surrounding a boundary or figure, activate the cursor, position it outside the boundary, select a color, and depress I. All unbounded areas will be colored by the seed (or Internal) fill algorithm incorporated in the AED 512/767 firmware.

Depressing RESET twice will clear the screen, returning the terminal to its initialized condition.



 RED

BLUE 

KEYSTROKE SEQUENCE
FOR RED BOUNDARY FILL
(CURSOR AT P1 OR P2)

B
CONTROL A

KEYSTROKE SEQUENCE
FOR BLUE BOUNDARY FILL
(CURSOR AT P2 OR P3)

B
CONTROL D

Figure 3.3
Boundary Fill

Intersecting circles or polygons can be colored using V (oVerlay fill) instead of I (Interior fill), the result being a Venn diagram with primary colors additive in common bounded areas. Refer to the IFL and OFL commands.

Figure 3.3 illustrates use of the boundary fill capability of the terminal. Refer to the BFL command.

To change the entire screen to a color other than black, command the terminal to: 1) change background color, 2) select the color, and 3) erase screen and print it in the newly selected color. The keystroke sequence would be (to get a white screen) —

KEYSTROKE COMMENT

LOCAL

ESC	Puts terminal in Interpreter Mode
[Left bracket, single character code for “set background color”
CTRL G	Says color desired is white (CTRL G = 7 in ASCII code)
CTRL L	Form feed; clear screen and advance to new page/screenful, with newly assigned color.

Locating the cursor at an intersection of lines or at any other area of interest, then depressing ZOOM IN, *centers the area of interest and magnifies it* by 2. Repeatedly depressing ZOOM IN produces successive integer magnifications (X2, X3, X4, , X16) up to 16 times. Depressing ZOOM OUT decreases the magnification by integers (x15, x14, x13, x12, x1). A very important feature of the terminal is the ability to select some particular point (and surrounding area) from any place on the video display and, on command, have the monitor re-draw the display with the selected point and region magnified and moved to the center of the screen. This is accomplished by depressing CURSOR, then manipulating the joystick to place the cursor on the point of interest. Depressing ZOOM IN will center the cursor and magnify the image by increasing integers.

If the cursor is enabled, the Pixel under the cursor will remain stationary on the screen, with the rest of the image expanding away from the first point as ZOOM IN is successively depressed. **If the cursor is disabled, the upper left corner of the screen will remain stationary; the image expands down, and to the right (AED 767 only).**

To move the display about the monitor screen, depress PAN and move the joystick. Switch the joystick to the *rate mode* (velocity and direction proportional to joystick deflection) by depressing the RATE key adjacent to the CURSOR key. Move the joystick to various angles and observe the screen images or cursor.

3.5 COMMUNICATIONS

The AED 512/767 communicates with computers, plotters, digitizer tablets, modems, etc., through the connectors shown in Figure 3.2. Select the proper cable(s) and interconnect the AED 512/767 terminal to your CPU, modem, etc. For serial interfaces, be sure the BAUD RATE is set properly because *experience has shown this to be a common source of installation difficulty.*

3.6 SOFTWARE

HOST computer software, which may include a Terminal Access Package (TAP), permits the AED 512/767 to operate as a teletype or video display unit utilizing bit serial transfers from 300 to 19,200 baud. The AED 512/767 emulates the Tektronix 4010-4014 series terminals and is, therefore, compatible with host software written for them. Direct Memory Access (DMA) transfers are achieved through the PARALLEL INTERFACE port and a CPU/DMA interface; however, a software device driver must, in some cases, be installed in the host software operating system.

3.7 OVERNIGHT OR CONTINUOUS USE

W A R N I N G

The AED 512/767 can be left with Power ON continuously; however, it is possible to damage the phosphorluminescent materials if one image is left on the screen for prolonged periods of time. To avoid this, either a) hit CTRL L (form feed) in Local Mode to clear the screen, b) activate RATE panning, or c) turn the unit OFF.

3.8 SWITCH SETTINGS: AED 512/767

COMMUNICATIONS SWITCH SETTINGS

Switch #	Function
1	X On/X Off Enable When Up
2	Locks RTS True When Up
3, 4, 5	Aux Baud Rate
6, 7, 8	Host Baud Rate

Setting Baud Rate: During system reset the baud rate is selected according to the back panel DIP switch labeled "COMM". This baud rate can be overridden from the keyboard or under host program control using the SBR function (See Terminal Command Protocol, Chapter 5).

- Positions 3 through 5 specify "AUX" baud rate.
- Positions 6 through 8 (right-most) specify "HOST" baud rate.

Comm. Switch Position			Communication
3	4	5 (AUX)	Rate
6	7	8 (HOST)	
DN	DN	DN	300 Baud
DN	DN	UP	600 Baud
DN	UP	DN	1200 Baud
DN	UP	DN UP	2400 Baud
UP	DN	UP	4800 Baud
UP	DN	UP	9600 Baud
UP	UP	DN	19200 Baud
UP	UP	UP	External Clock/16 *

* Baud rate of 9600; external clock should be $16 \times 9600 = 153.6$ KHz, for example.

OPTION SWITCH SETTINGS

(for Communications Protocol, Host to Terminal)

Switch #	Position	Result
1	UP	Half duplex transmission with local echo
	DN	Full duplex transmission.
2	UP	SPACE will erase characters spaced over. Backspace will not erase.
	DN	Backspace will erase character and fill background color.
3	UP	Rubouts are ignored
	DN	Rubouts are not ignored
4	UP	Priority given to TEK4014 command set. Always set this switch UP when emulating the 4014. The OPT command #6 overrides this switch when such emulation is not desired
	DN	Priority given to AED 512/ 767 command set
5	UP	Parallel: Full reset of terminal occurs when Command Out word, with bit 8 set, is transmitted. Serial: Standard bit stream configuration, only. See chart below.
	DN	Parallel: Terminal reset is independent of bit 8 of Command Out. Serial: With switches 6 - 8, specifies the serial data configuration as detailed in the following table.
6, 7, 8	Value 0 - 7 with MSB = Bit 6.	Parallel: With Switch 5 UP, specifies the device address among multiple terminals on I/O bus. Serial: With Switch 5 DOWN, specifies the serial data configuration as detailed in the following table.

Should it be necessary to change to another configuration of the "HOST" serial port, set the OPTIONS switch, positions 5 - 8 as follows:

Option	Switch Position				Character Configuration
	5	6	7	8	
UP	x	x	x	x	8 + 0 + 2 (Standard)
DN	DN	DN	DN	DN	8 + 0 + 2
DN	DN	DN	UP	UP	8 + 0 + 1
DN	DN	UP	DN	DN	8 + EVEN + 1
DN	DN	UP	UP	UP	8 + ODD + 1
DN	UP	DN	UP	UP	7 + EVEN + 2
DN	UP	DN	UP	UP	7 + ODD + 2
DN	UP	UP	DN	DN	7 + EVEN + 1
DN	UP	UP	UP	UP	7 + ODD + 1

NOTE: x = Does not matter

The standard serial character configuration is 1 start bit, followed by 8 data bits followed by 2 stop bits. This is abbreviated as 8 + 0 + 2, for:

- 8 Data Bits
- 0 Parity Bits
- 2 Stop Bits

NOTES:

1. Please use 8 data bits when possible.
2. If using multiple terminal parallel communication (below), it is not possible to depart from the standard serial configuration through DIP switch settings since positions 6 - 8 are used to select the terminal device number ("unit" numbered)

Because of the expanded command set on the AED 767, the full 8 bits transmitted from the host will now be recognized by the AED 767. On the AED 512, the parity bit is discarded by the serial interface and therefore the terminal can be run in "no parity" mode, regardless of the parity of the host computer. What this means is on the AED 767 the parity setting of the terminal must match the parity setting on the host or errors in transmission over the serial interface will occur.

Unlike the AED 512, the AED 767 has 3 sets of DIP switches on the back. The first 2 sets, COMMUNICATIONS and OPTIONS, have the exact same meaning as with the AED 512. The 3rd set, labeled DIS for DISPLAY, are used to change the screen resolution and image position. There are 8 dip switches and the settings are as follows:

DISPLAY SWITCH SETTINGS

Position Of Switch	#1	#2	Result
	DN	DN	Display window is fully to the left in video memory.
	UP	DN	Display window is 32 pixels to the right of the above.
	DN	UP	Display window is 64 pixels to the right.
	UP	UP	Display window is 96 pixels to the right.
Switch	#4	#5	
	DN	DN	Setting NOT ALLOWABLE*
	UP	DN	Visible window is 768 pixels wide
	DN	UP	Visible window is 640 pixels wide.
	UP	UP	Visible window is 512 pixels wide.
Switch	#6	#7	
	DN	DN	High refresh rate (37 to 41 Hz)
	UP	DN	Setting NOT ALLOWABLE*
	DN	UP	Setting NOT ALLOWABLE*
	UP	UP	Low refresh rate (24 to 31 Hz)
Switch	#8		
	DN		Visible window is 483 lines high.
	UP		Visible window is 575 lines high.

***CAUTION:** Setting Switches 4 and 5 or 6 and 7 to these *not allowable* combinations will cause unpredictable results and may damage both the base unit and monitor. AED will not be responsible for any damage caused by incorrectly setting these switches. Change and verify switch settings with power OFF.

The AED 512/767 can refresh at 2:1 interlace compared to power line frequency. For example, with 60Hz power either terminal will refresh all scan lines on the screen in 1/30th of a second, the same condition which exists with USA broadcast television receivers. Single horizontal vectors and, to an extent, pixel matrix text characters will appear to flicker when displayed on standard persistence monitors. Again, the same is true with standard broadcast TV receivers. At 50Hz power the flicker is worse because the 2:1 interlace results in a 1/25th of a second refresh period. To reduce the flicker problem, long persistence monitors can be used; so the system still conforms to broadcast TV refresh standards - NTSC or CCIR. Unfortunately, long persistence phosphors smear the image when it is dynamic.

To avoid having the refresh rate inflexibly set at twice the power line frequency, the raster scan timing can be synchronized to an internally generated, stable clock. This happens when the Video Synchronizing Switch in Figure 3.1 is set to INternal. Then it is very possible to have, for example, a 30 Hz refresh rate (rather than 25 Hz) when operating the terminal and monitor from 50 Hz power. The combination of going to internal sync and providing both NTSC and CCIR sync PROMS provides an alternative which has been incorporated in the AED 767-switch selectable, higher refresh rates which result in considerably less flicker with standard persistence monitors. Because the number of pixels per second which can be delivered to the screen is limited, refresh rate and number of pixels per screen (resolution) are interrelated.

The AED 767 should be set to the following combinations; others produce less satisfactory image quality and require extensive internal adjustment.

DISPLAY WINDOW/REFRESH RATE SELECTION

#4	Display Switch Settings				Display Window	Refresh Rate	Sync Switch Setting
	#5	#6	#7	#8			
UP	DN	DN	DN	UP	768H X 575V*	37Hz Approx.	INternal
DN	UP	DN	DN	DN	640H x 483V	42Hz Approx.	INternal
UP	UP	UP	UP	DN	512H x 483V	30Hz	LINE
DN	UP	UP	UP	DN	640 x 483V**	30Hz	LINE
UP	UP	UP	UP	DN	512H x 483V	25Hz	LINE***

* Standard configuration shipped. If terminal is supplied with monitor, both will be adjusted for 37 Hz refresh.

** May require internal adjustment to terminal

*** With 50 Hz power, a 30 Hz refresh rate may be obtained with an NTSC sync PROM (switches 6 and 7 UP) and the sync switch to INternal.

Because AED supplied monitors are factory adjusted to a 37 Hz refresh rate, it is likely that use at 25 or 30 Hz will require monitor adjustment.

3.9 GENERAL INSTALLATION CHECK LIST FOR AED 512/767 AND HOST CPU

1. _____ Check line voltage - - 240V OR 110V
2. _____ Check fuse flip chip (if on your terminal).
3. _____ All cable connections solid; arrows matched, pin 1 to pin 1; cable in good condition.
4. _____ Boards in CPU seated correctly.
5. _____ Check baud rate, communication switch, parity.
6. _____ Power on everything, CPU last!

3.9.1 Troubleshooting

PROBLEM	CHECK
No chirp, no response (dead unit)	Unit plugged in? Power LED on? Fuse OK? Repeat power up sequence.
Unit chirps - - no signature on screen e.g. AED 512B V82.9	RGB cables connected? Power cable for monitor plugged in? Monitor on/off switch activated? Toggle switch on terminal set to line or internal position?
Excessive flicker	Adjust brightness and gain
Monitor displays "check sum error"	Indicates bad PROMS-replace.
Vertical lines on power up or fill	Probable bad memory chip.
Plaid pattern or individual pixels displayed	Component failure - e.g.: bad chip, PROM, keyboard.
Unit powers up but keyboard will not respond	Remove terminal top cover and check for cable connections.

PROBLEM

CHECK

All LED's on and/or indistinguishable patterns on screen.

Improper power-up sequence, power-up CPU last.

Reset, turn power OFF and ON.

Some component is "hanging" bus.

*Disconnect communication cables and try 512/767 in stand-alone mode.

512/767 chirps repeatedly

Buffer overload. If operating with CPU, is COMM switch #1 up enabling X ON/ X OFF? If in stand-alone mode, a component failure is indicated.

Patches of uneven color on screen

Purity problem. Degauss. Fill screen with red. Press degaussing button on back of monitor and hold in for several seconds. Monitor should be warmed up for approximately 10 minutes before degaussing.

Do not store metal objects on top of monitor

767 will not respond to 3 character commands

767 defaults to lower case. CAPS LOCK key must be enabled to execute 3 character mnemonics.

Tektronix emulation won't operate

Option switch 4 up? Check baud rate and parity.

Graphic tablet will not function

Cursor on? ETC command issued? Aux baud rate same as bit pad baud rate? Degauss bit pad with magnet. Pen and power supply properly connected?

RS232 cable-connections to pins 2 and 3 crossed on one end of cable?

DIP switches inside bit pad set properly?

Daisy chain of monitors not functioning properly or image is too dark in hue

19" monitor must be last in chain. Set 19" monitor termination switch to 75 ohms. 13" monitor should be set to high impedance.

*If the 512 or 767 functions in stand-alone mode, re-connect RS232 cable to the terminal. Remove all I/O devices from the CPU, leaving in the CPU card, memory, serial interface and terminator. (CAUTION: Power down CPU before removing boards. THIS IS ESSENTIAL!!!) Hit break key to test for serial communication. If there is no serial communication, suspect a computer problem, bad RS232 cable, or possible serial port failure. First try another cable. Then try removing boards and cleaning the gold part of connector fingers with an eraser. Carefully reset boards and try again. If serial communication works start adding I/O devices one at a time, the DMA interface first. Check again for serial communication. Keep adding I/O devices until you find the problem device or cable.

NOTE: The flat gray I/O cable must be connected properly to the unit and I/O device during testing for parallel communication .

4. KEYS AND INDICATORS

4.1 SPECIAL FUNCTION KEYS (4-1)

4.2 STANDARD FUNCTION KEYS (4-8)

4.3 INDICATORS (4-8)

4.4 THE NORMAL (7X12) FONT (4-10)

— ASCII CODE CHARTS

4 KEYS AND INDICATORS

This chapter describes the keys and indicator lights on the AED 512/767 keyboard. The user is assumed to have read *Chapter 3* entitled *Installation*, and to have performed the suggested keyboard operations described there. In this chapter, *a more complete description of each key's function* is provided. Please refer first to Figure 4.1A or 4.1B, following, to refresh your memory of the location of the keys on your terminal. Perhaps even better, you should sit down at your AED 512 or AED 767 and try out the keys as we describe them. The keys are treated from the top, left to right on your keyboard.

4.1 SPECIAL FUNCTION KEYS

Reset

Hitting the RESET key the first time interrupts the terminal's current task and causes it to enter *Interpreter Mode*. (If already in the Interpreter Mode, the first RESET cancels a pending or partially completed command.) Hitting this key again after one second (without any other keystroke) invokes a *FULL RESET* which reinitializes the terminal. All default modes and conditions (such as Baud rates on HOST and AUX ports) are reinstated, self-test diagnostics are performed, and the terminal is placed in *Alphanumeric Mode*. Following initialization, the terminal signature appears on the upper left corner of the display and then the terminal chirps, signifying that no self-test errors have been found.

Local

After initialization (FULL RESET) the terminal is in *Remote Mode*. In this mode, data can be exchanged with the host through the serial or parallel communication ports. Pressing the LOCAL key puts the terminal in *Local Mode*. In Local Mode, keystrokes entered on the terminal are moved to the receive buffer where they are processed as if they had come from the host. Data sent serially by the host is ignored; host data sent via parallel transmission is not.

Lower Case (AED 512 only)

After the AED 512 has been initialized, it is in *ALL CAPS Mode*. To generate lower case codes, hit the LOWER CASE key. The indicator light above the key will light up showing that Lower Case Mode is in effect. Use the SHIFT key to generate capital letters while in *Lower Case Mode*.

Caps Lock (AED 767 Only)

When the AED 767 is initialized, the keys are in *Lower Case Mode*. Therefore, if you are in *Local Mode*, you must press the CAPS LOCK key or depress the SHIFT key before you can execute certain commands. Pressing the CAPS LOCK key puts the terminal in *Upper Case Mode* until *FULL RESET* or again depressing CAPS LOCK.

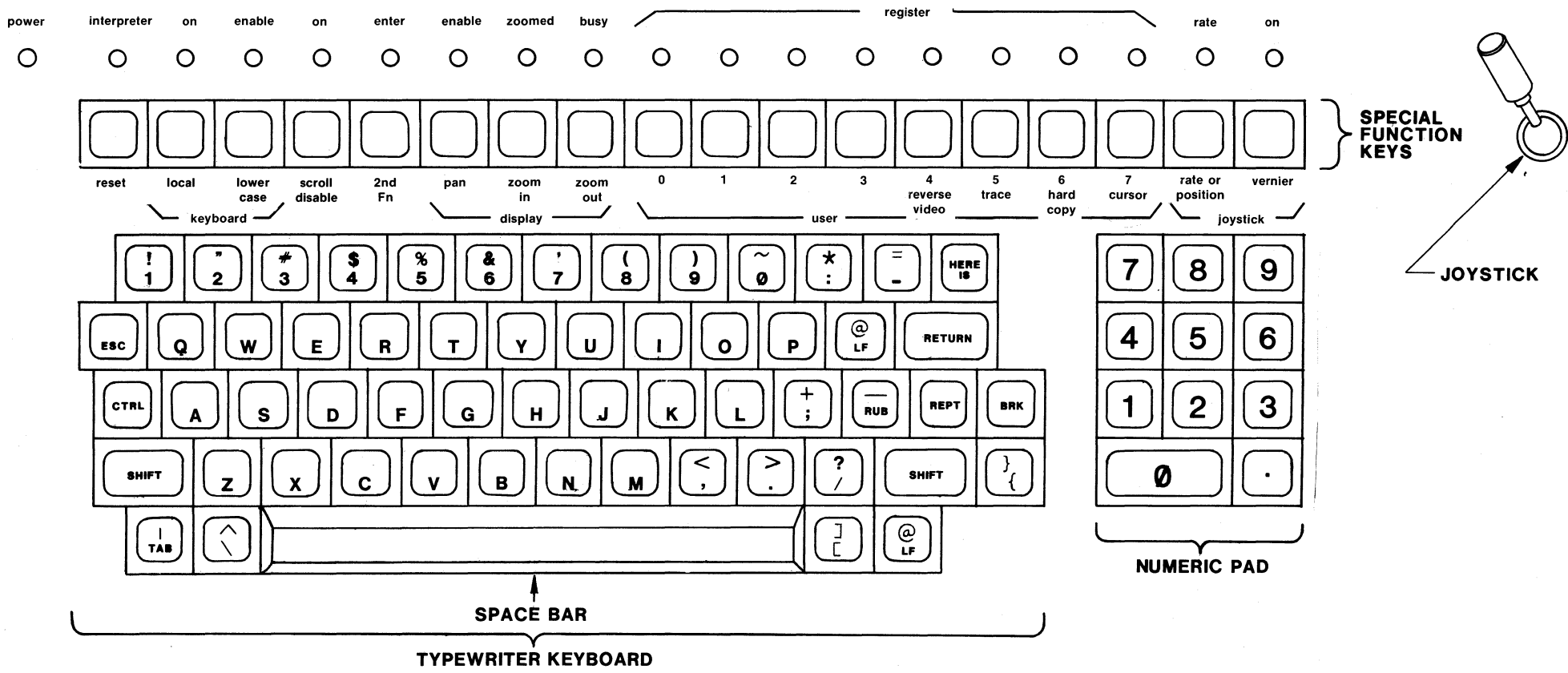


Figure 4.1.A
AED 512 Keyboard

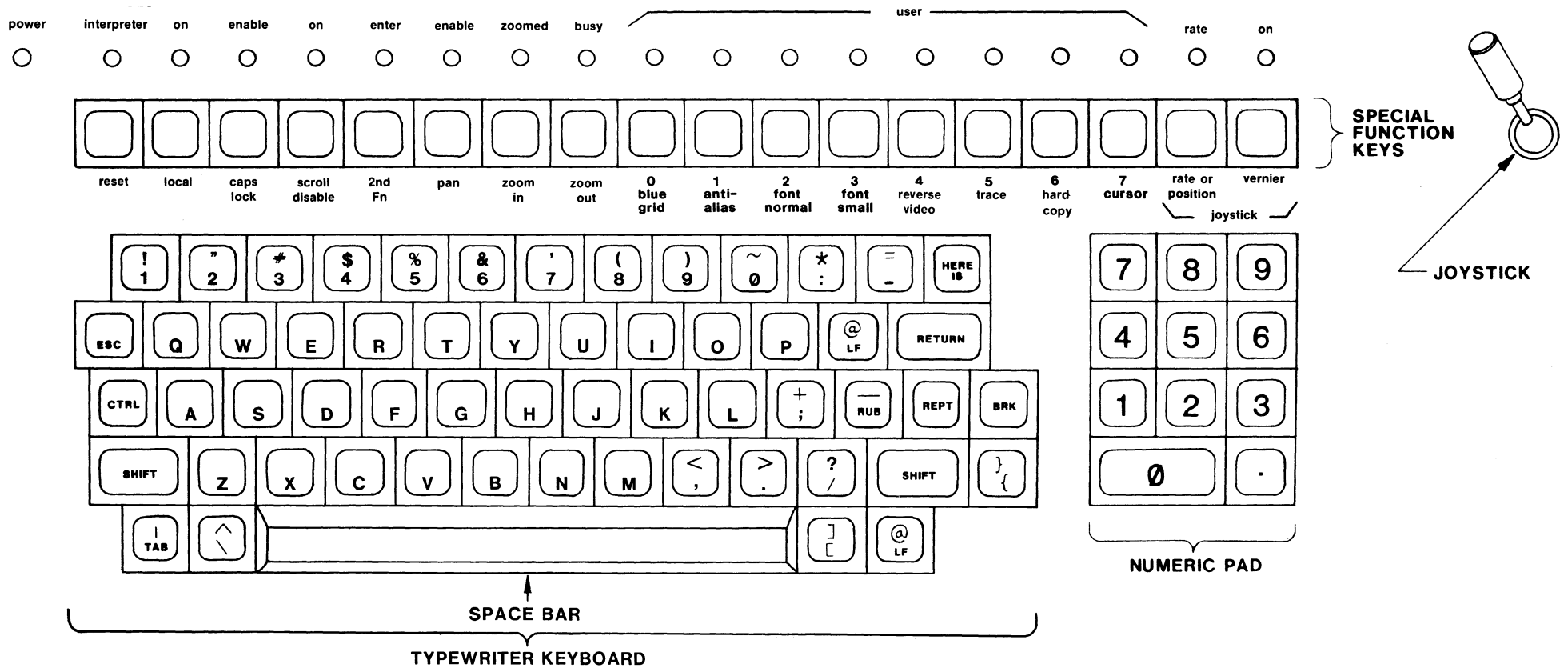


Figure 4.1.B
AED 767 Keyboard

4.1 SPECIAL FUNCTION KEYS (continued)

Scroll Disable

Normally, when scrolling begins, after the bottom line of text has been written, the entire image is scrolled (or moved) upward by one line. Then the next line of text is written into the available new bottom line. The top line of text is moved off-screen and erased from video memory.

However, if the SCROLL DISABLE key has been pressed, the terminal will stop scrolling (hang) after the bottom line has been filled. The Scroll Disable indicator will light and no additional characters will be displayed. This feature "freezes" the display to allow the reading of text that otherwise would be scrolled upward too rapidly to be read. Depressing SCROLL DISABLE *again* will reinstate scrolling.

A Form Feed (CTRL L) character code generated while text is being sent will also stop scrolling of the text. CAUTION: If the host computer is communicating over the serial interface and does not support XON/XOFF or RTS handshaking, the incoming text may be lost.

Pan

Depressing this key lights the corresponding indicator and places the terminal in *Joystick Pan Mode*. Absolute or Rate pan can be selected with the RATE key (described below). *Pan Mode* will remain in effect until the PAN key is hit once more or the Disable Pan (DPA) command is issued. The joystick may be used to move the display in the X-Y plane.

If the Cursor Indicator and the Pan Indicator are both lit, the cursor pans with the screen image as the joystick is moved.

The term panning is used to refer to movement of the display window in X or Y, or both, coordinates. It thus includes scrolling as the special case of vertical (Y direction) movement.

Zoom In

On the AED 512, when the ZOOM IN key is depressed *with the cursor OFF*, the image at the center of the screen will remain in the center with a X2 magnification. Depressing the ZOOM IN key again will magnify the same image to X3, etc. *If the cursor is ON*, the image area it covers will be moved to the center of the screen and magnified, but the cursor will remain in the same physical position on the screen where it was just before the ZOOM IN key was depressed. Thus, depressing the ZOOM IN key again will move to the center of the screen whatever area of the image is covered by the cursor. The image will keep moving as the ZOOM IN key is depressed. To prevent the image from successively moving, either reposition the cursor to the image area of interest or extinguish the cursor after the first zoom.

On the AED 767 *with the cursor OFF*, the zoom is centered in the *upper left corner* of the screen, so areas to the right and below the corner object will continue to move to the right and toward the bottom of the screen as the ZOOM IN key is successively depressed. *With the cursor ON*, the image under the cursor will remain at that screen location as the ZOOM IN key is successively depressed; that is, *the zoom is centered on the cursor*.

4.1 SPECIAL FUNCTION KEYS (continued)

Zoom Out

Depressing this key will contract (demagnify) the display to the next lower zoom factor. When zoom reaches X1, the Zoomed Indicator goes out. The current screen center will not change during ZOOM OUT, whether or not the cursor is ON.

User Programmable Keys, 0 Through 7

Keys 0-7 are programmable by the user with commands DPK and DSK. Therefore, they are "soft" keys. The AED 512 and AED 767 keyboards have some differences in the default uses of these keys as shown below:

	Default (Unprogrammed)	
Key	AED 512	AED 767
0	—	Blue Grid
1	—	Anti-alias
2	—	Font Normal
3	Font Small	Font Small
4	Reverse Video	Reverse Video
5	Trace	Trace
6	Hard Copy	Hard Copy
7	Cursor	Cursor

The functions listed above and *described below* are in effect after power-up or after a full reset; they persist until programmed by the terminal user.

USER PROGRAMMABLE SPECIAL FUNCTION KEYS

Anti-Alias

Depressing this key invokes the anti-aliasing function which causes subsequent vectors drawn with colors 0 through 7 to be visually smoothed. Depressing this key a second time causes subsequent vector drawing of colors 0 through 7 to become jagged again. See Chapter 1 and the Anti-Alias Vector (AAV) command in Chapter 5.

Blue Grid

Depressing this key causes a rectangular grid to be nondestructively superimposed on any image just previously displayed. The grid has blue lines at 8, 32 and 64 pixel spacing (horizontal and vertical) with intensity as a function of the spacing. Depressing this key again turns OFF the blue grid. See the BLG command in Chapter 5.

Font Normal

After FULL RESET the terminal (normal) text characters are in a matrix 7 pixels wide by 12 pixels high, with 8 pixel horizontal spacing and 13 pixel vertical spacing between centers. See Section 4.4. Should an alternative font have been selected, depressing this key will return the font to the normal size.

4.1 SPECIAL FUNCTION KEYS (continued)

Font Small

Depressing this key will select a font which has characters in a matrix 5 pixels wide by 7 pixels high, with 6 pixels horizontal and 8 pixels vertical spacing.

Reverse Video

Normally, the monochrome output will be a white-on-black rendering of data *in the least significant memory plane*. Toggling this key will result in a black-on-white, reverse video display.

Trace

This key suppresses the incoming escape (ESC) character and prevents the terminal from operating in the Interpreter Mode. Thus, *the incoming printable characters are printed on the screen* rather than being interpreted as commands or arguments. This is a debugging aid when the terminal is connected to a computer.

Hard Copy

If in *Alpha Mode*, data is copied to the auxiliary serial port (AED 512 only). AED 767 data is copied either to the auxiliary or the parallel port, depending on Option #7.

Cursor

Depressing this key causes the cursor to be displayed on the screen and allows the cursor to be moved with the joystick. With the cursor OFF, the Draw Cursor Absolute (DCA) command will cause the cursor to appear at the CAP. However, *if the cursor is ON* or the Enable Joystick Cursor (EJC) command is invoked, the *cursor will follow the joystick position* rather than the CAP unless the CAP happens to coincide with the joystick position.

Rate

This key enables/disables the *Joystick Rate Mode*. In *Rate Mode*, horizontal and vertical image displacements will vary at a *rate* proportional to the joystick deflections from center position. A dead zone is provided in the center to allow freezing of the joystick values. When the joystick is not in *Rate Mode*, it is in *Position Mode*, where H and V components of *image displacement* depend only upon the *absolute joystick position*.

Vernier

This key enables joystick 16 to 1 vernier position control, i.e., improves (decreases) the sensitivity of cursor position to joystick movement.

HERE IS

No official current function. Pushing HERE IS may clear the screen without invoking RESET on your terminal, but this may change in future firmware.

4.1 SPECIAL FUNCTION KEYS (continued)

ESC

Sends ASCII ESC (decimal 27), and, if in *Local Mode*, lights the Interpreter Indicator.

LF (Line Feed)

Causes alphanumeric cursor (underline) to move one line space downward for each key depression. No horizontal movement results.

RETURN

Moves the alphanumeric cursor to the left margin of the display. No vertical movement results.

CTRL (Control)

Used in conjunction with another key to generate a non-printing character for control purposes. For example, if the terminal is in single character function code encoding mode, coincident depression of CTRL and L keys generates a "Form Feed (FF)" command, 12 decimal or 0001100 binary, which clears the old display page to the background color but does not invoke RESET. See Figure 4.3.

RUB (Rub Out)

Sends ASCII DEL, 127 decimal. Most operating systems recognize this as "ignore last keystroke." In the *Local and Alphanumeric (Text) Mode*, depressing RUB OUT will erase the last typed character.

The RUB OUT command (from Host or AED keyboard) causes a cell to be written in the background color, backspaced one cell along the horizontal axis of the screen. The default cell size is 8H X 13V pixels and the background color is black, thus any selected character or image area of the cell size will be obliterated.

REPT (Repeat)

Repeats the last key typed, at a rate of approximately ten characters per second. If a programmable function key was the last key depressed before REPT is depressed, then the contents of that programmable key will be repeated. This is a useful feature if the programmable key contains the instructions to draw a circle, draw a filled circle, or fill a rectangle relative (FRR, Chapter 5) because the *joystick (or tablet stylus) can be used as a paint brush* while the REPT key is held down.

BRK (Break)

If the BREAK and CTRL keys are simultaneously held down, the terminal will send a break character over the HOST serial line. This is essentially a null character with no "stop" bits affixed. Depressing BRK by itself has no effect.

SHIFT

Use the SHIFT key to *generate capital letters* while the terminal is in *Lower Case Mode*. See Lower Case and Caps Lock headings above.

4.1 SPECIAL FUNCTION KEYS (continued)

SPACE BAR

In *Alphanumeric Mode*, the space bar causes the alpha cursor to *move one letter space to the right*. When the right page border is reached the terminal automatically generates a carriage return and a line feed.

NUMERIC PAD

Duplicates the digits 0-9 and a decimal point, for convenience.

TAB

Same as space bar in *Local Mode*. If in *Interpreter Mode*, TAB returns the terminal to the Alpha Mode. On keyboards without TAB key, hit CTRL I to generate the responses just described. The terminal firmware is not programmed to provide a typewriter-like multiple space jump by the CAP (Current Access Position).

Nul

To generate a character with binary value 0 (nul), hit the two keys CTRL @ simultaneously. Not a key, but an occasionally needed character.

4.2 STANDARD FUNCTION KEYS

Refer to the Keyboard drawing, Figures 4.1 A and B, and the ASCII Character Code Chart, Figures 4.3 A and B, for information concerning the additional control functions normally provided on exclusively alphanumeric terminals. For example, Form Feed = CTRL L. Of course the letter and decimal digit keys are standard function keys.

4.3 INDICATORS

From left to right, above the top row of the keyboard are the indicator lights. The label and function (when lit) of the indicator lights (LED's) are defined as follows.

Power

When lighted, indicates presence of the +5 volts.

Interpreter

Indicates that the terminal is in *Interpreter Mode* (not *Alphanumeric*). Incoming characters are not printed but are interpreted as a command function code and arguments.

On (Local Mode)

Indicates terminal is in *Local Mode*. Note that indicator is located just above the LOCAL key.

Enable (Lower or Upper Case)

Indicates that *Lower Case Mode* is enabled (AED 512). Indicates that *Upper Case Mode* is enabled (AED 767).

On (Scroll Disable)

Indicates that the SCROLL DISABLE key, located just below the indicator, has been depressed, and if X ON/X OFF is invoked, incoming characters from the host have been stopped. If X/ON/X/OFF has *not* been invoked, incoming characters are being lost once the 256 byte character buffer is full.

Enter

No current function.

Enable (Pan)

Indicates either that the PAN key has been depressed, or that an Enable Pan (EPA) instruction code has been received. Note the indicator is just above the PAN key.

Zoomed

Indicates that the terminal is zoomed greater than X 1.

Busy (Interpreter)

Indicates the interpreter is executing a function code or awaiting the completion of a keystroke sequence.

Register

These eight indicators may be lit by the DRL command. The command's argument will light a pattern of indicators based on a binary coded decimal equivalent. For example: DRL1. will light the far right indicator (above USER KEY 7); DRL2. will light the indicator second from the far right (above USER KEY 6); DRL 255. will light all eight indicators; and DRL0. will extinguish all indicators.

If the DRL command is not invoked but a corresponding user function key is, the indicator above that key will light. (See the above section on USER PROGRAMMABLE SPECIAL FUNCTION KEYS 0-7) If the DRL0. command is issued the light will be extinguished even though the corresponding function key is still active.

Rate

Indicates that the joystick "RATE" or POSITION" key (just below the indicator) has been depressed, subsequent to the last initialization. Cursor movement (rate) will now be proportional to joystick deflection from a central position.

On (Vernier)

Indicates that the VERNIER key just below the indicator has been depressed, subsequent to the last initialization. The sensitivity of cursor movement to joystick displacement is now reduced, allowing vernier (fine) control.

4.4 THE NORMAL (7 X 12) FONT

Figure 4.2 shows sample pixel patterns (icons) as well as the location of the current access position (CAP) relative to the character body. Each character is located within a 7 x 12 matrix. The 7 x 12 matrix is contained in an 8 x 13 pixel cell which guarantees proper spacing between characters.

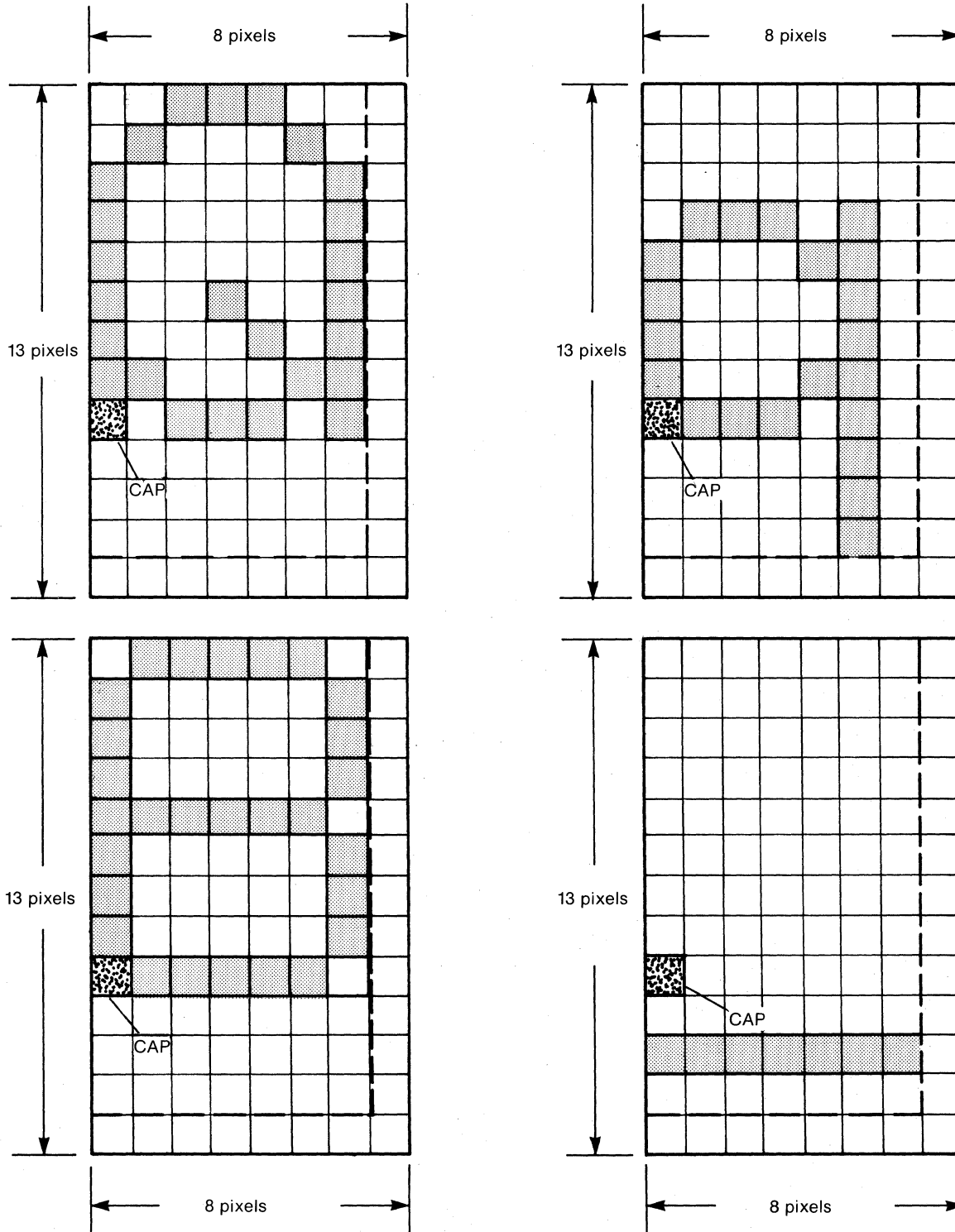


Figure 4.2
Normal 7 x 12 Font

ASCII CODE CHART

SEVEN UNIT CODES, ASCII *

OCTAL			OCTAL	
** 000	NUL	(Blank)	100	@ (Ampersand)
001	SOH	(Start of Header)	101	A
002	STX	(Start of Text)	102	B
003	ETX	(End of Text)	103	C
004	EOT	(End of Transmission)	104	D
005	ENQ	(Enquiry)	105	E
006	ACK	(Acknowledge (Positive))	106	F
007	BEL	(Bell)	107	G
010	BS	(Backspace)	110	H
011	HT	(Horizontal Tabulation)	111	I
012	LF	(Line Feed)	112	J
013	VT	(Vertical Tabulation)	113	K
014	FF	(Form Feed)	114	L
015	CR	(Carriage Return)	115	M
016	SO	(Shift Out)	116	N
017	SI	(Shift In)	117	O
020	DLE	(Data Link Escape)	120	P
021	DC1	(Device Control 1)	121	Q
022	DC2	(Device Control 2)	122	R
023	DC3	(Device Control 3)	123	S
024	DC4	(Device Control 4)	124	T
025	NAK	(Negative Acknowledge)	125	U
026	SYN	(Synchronization)	126	V
027	ETB	(End of Text Block)	127	W
030	CAN	(Cancel)	130	X
031	EM	(End of Medium)	131	Y
032	SUB	(Substitute)	132	Z
033	ESC	(Escape)	133	[(Opening Bracket)
034	FS	(File Separator)	134	\ (Reverse Slash)
035	GS	(Group Separator)	135] (Closing Bracket)
036	RS	(Record Separator)	136	^ (Circumflex)
** 037	US	(Unit Separator)	137	_ (Underline)
040	SP	(Space)	140	' (Open Single Quote)
041	!	(Exclamation)	141	a
042	"	(Quotation)	142	b
043	#	(Number Sign)	143	c
044	\$	(Dollar)	144	d
045	%	(Percent)	145	e
046	&	(And Sign)	146	f
047	'	(Closing Single Quote)	147	g
050	((Left Paren)	150	h
051)	(Right Paren)	151	i
052	*	(Asterisk)	152	j
053	+	(Plus)	153	k
054	,	(Comma)	154	l
055	-	(Hyphen)	155	m
056	.	(Period)	156	n
057	/	(Slash)	157	o
060	0		160	p
061	1		161	q
062	2		162	r
063	3		163	s
064	4		164	t
065	5		165	u
066	6		166	v
067	7		167	w
070	8		170	x
071	9		171	y
072	:	(Colon)	172	z
073	;	(Semi-colon)	173	{ (Opening Brace)
074	<	(Less Than)	174	(Vertical Line)
075	=	(Equal)	175	} (Closing Brace)
076	>	(Greater Than)	176	~ (Overline (Tilde))
077	?	(Question Mark)	177	DEL (Delete/Rubout)

* This code is often sent with a parity bit and is thus sometimes referred to as an eight-bit code.

** The codes from contiguous octal values 000 through 037 are non-printing control characters.

Figure 4.3.B
ASCII Code Chart

5. TERMINAL CONTROL PROTOCOL (TCP)

5.1 INTRODUCTION (5-1)

5.2 FUNCTIONAL INDEX OF COMMANDS (5-2)

5.3 ALPHABETICAL ORDERING OF COMMANDS (5-4)

5. TERMINAL COMMAND PROTOCOL

5.1 INTRODUCTION

TERMINAL CONTROL STATES

During power-up or manual reset, the terminal enters Alphanumeric Mode. In this state it acts in a fashion usual to alphanumeric-only terminals; that is, characters received are displayed on the screen from left to right, top to bottom. The normal ASCII control codes (carriage return, line feed, form feed) are accepted and appropriately processed.

Upon receipt of ESC code (27), the terminal enters graphic Interpreter Mode. Rather than being displayed, the next character received is interpreted as a "function code". Depending on the definition of the particular function, it may have one or more arguments ("operands"). The host program must send (or in Local operation, the operator must enter) the expected number of operands because the next character will be interpreted as new function code. The interpreter returns to Alphanumeric Mode when a function code received is either:

1. A control character other than ESC(27), GS(29), or nul(0).
2. An undefined function.

This character is then processed in Alphanumeric Mode.

The preceding paragraph describes the default encoding scheme for doing graphics in the Interpreter Mode. In addition to this Single Character, Non-Mnemonic function coding, there is a Three Character, Mnemonic coding which provides an alternate way to invoke interpreter functions. For key graphics data entry, the operator can use the longer but more easily remembered codes (e.g. set color, SEC) as opposed to the arbitrary single-character codes (e.g. set color, C).

FUNCTION AN OPERAND ENCODING

All function codes have definitions which are printing characters in ASCII. In the description that follows, function codes are given in all caps followed by necessary operands, if any. The actual characters or bytes expected by the interpreter depend upon the encoding scheme specified with the SEN function. Normally the function code will be one character and each operand one or more 8-bit bytes, which can be tailored to the user's requirements. For instance, when a host cannot send certain control characters, the user might select ASCII hexadecimal encoded operands.

FUNCTIONAL INDEX OF COMMANDS

ALPHANUMERIC

Text Characters - HOME, Set Alpha cursor Color, Select Alpha Parameters, Set Console Status, XXX-eXit graphiX, enter teXt mode

COLOR CONTROL

Color Input - Load Anti-alias Table, Set Background Color, Set Alpha cursor Color, Set BLink, Set Cursor Colors, Set Color Table, Write Horizontal sCan-not AOI, Write Horizontal Scan - AOI, Write Horizontal Runs, Write Horizontal rUns, Write Pixel

Color Output - Read Pixel, Read Color Table, Read Horizontal Runs, Read Horizontal Scan

Color Selection - Set Current Colors

COMMUNICATIONS

Direct Memory Access - Define Area of Interest, Read Direct from AOI, Read Raster Direct, Start Command DMA, Set Console Status, Stop Direct Access - Read or Write, Select Interface - S or P, Set Up Counters - For DMA, Write Direct into AOI, Write Raster Direct, eXit Command DMA

Parallel - Set Console Status, Set Interface - S or P, Send Carriage Return, Stop Direct Access.

Run Length Encoding - Define Area of Interest, Read Horizontal Runs, Read Horizontal Scan, Write Horizontal sCan (not AOI) Write Horizontal Runs, Write Horizontal Scan, Write Horizontal rUn alternate, Write Incremental Plotter mode, Write Multiple Pixels

Serial - Copy to Aux, Programmable Option Number, Set Baud Rate, Send Carriage Return, Set Console Status, Set InterFace - S or P, Send Key Stroke, Set Turnaround Delay, Write Horizontal Runs

IMAGE PROCESSING

Run Length Encoding - Read Color Table, Read Horizontal Runs, Read Horizontal Scan, Write Direct into Area of interest, Write Horizontal sCan, Write Horizontal RUNs, Write Horizontal Runs alternate, Write Raster Direct

INTERPRETER

Graphics Mode - ESCape to interpreter, ReSet Terminal, Set ENcoding, XXX-eXit graphiX enter text mode

KEYBOARD AND INDICATORS

Function Keys - COpy, Define Programmable Key, Define Soft Keys, Set Console Status

Indicators - Define Register LED

MEMORY FEATURES

Firmware

User - Jump User Subroutine, Load Microprocessor Ram, Set Stack End

Video - ERaSe write enabled planes, Form FeeD, MOVe cap absolute, MoVe cap Relative, Read CaP, Read Direct from AOI, Read Horizontal Origin, Read Horizontal Scan, Read Vertical Origin, Set Read Mask, Set Vertical Origin, Set Write Mask, Write Direct into AOI, Write Horizontal Runs, Write Horizontal Scan, Write Horizontal RUN alternative

SUPERDAM

SUPERoam, Set Read Mask

TEKTRONIX

GS - Enter = Tektronix Mode, programmable OPTion number, Set Tektronix Window (767 only), SUB = enter Tektronix compatible mode.

5.2 ALPHABETICAL ORDERING OF COMMANDS

The remainder of this chapter contains a detailed description of each terminal command. Commands are ordered alphabetically, by their 3-character mnemonic names. In the following examples, all references to enabling the "CAPS LOCK" key refer to the AED 767 ONLY.

CURSOR, JOYSTICK, TABLET

Cursor Control - Draw Cursor Absolute, Erase Cursor Unconditionally,
Set Alpha cursor Color, Set Cursor Colors, Set Cursor Parameters,
Set Console Status

Cursor Output - Read CaP

Joystick - Disable Joystick Cursor, Enable Joystick Cursor, Read
Joystick Position

Tablet - Define Tablet Mapping, Enable Tablet Cursor, Enable Tablet
Polling, Read Tablet Position, Set Tablet Parameters

DISPLAY SCREEN CONTROL

Background Color - Blue Line Grid, Set Background Color, Set
Console Status

Scroll - Both Set Origins, Read Vertical Origin, Set Auto Roam,
Set Vertical Origin, Vertical Scroll Relative

Pan - Disable PAN, Enable PAN, Horizontal Scroll Relative, Read
Horizontal Origin, Set Auto Roam, Set Horizontal Origin

Zoom - Set Zoom Registers

GRAPHICS PRIMITIVES

Circles - Draw Circle, Draw Fat Circle

Ellipses - draw ELLiPses

Lines - Set Line Style

Pixels - Read PiXels, Write PiXel

Polygons - Boundary FiLL, Draw Filled Polygon, Generalized overlay
FiLL, Interior FiLL, Overlay FiLL

Rectangles - Draw Filled Rectangle, Define Stipple Pattern, Fill
Rectangle Relative, Set Pattern Fill

Special Font - Define Special Font, Erase Special Font, Write
Special Font

Vectors - Anti-Aliased Vectors, Draw Multiple Vectors, Draw Vector
Absolute, Draw Vector Relative, Load Anti-alias Table

LEGEND

1. CAP is the abbreviation for Current Access Position, AOI for Area of Interest.
2. All numbers are in decimal, except where otherwise noted.
3. Text in quotes is ASCII encoded; e.g., "A"=65.
4. The two-character sequence xx denotes a "don't care" bit.
5. Digit zero (0) will be slashed (0) only where necessary to prevent confusion. Note the difference between letter O and numeral 0.
6. The least significant bit in a given field is numbered 0.
7. Operand text in parentheses () is repeated one or more times.
8. Operands are 8-bits except where otherwise noted by a numeric suffix, e.g., n16 indicates 16 bits.
9. Operands preceded by a slash (/) are returned by the terminal to the host.
10. Command names will be spelled, occasionally, to emphasize the three character encoding of the commands. For example, Boundary FiLl (BFL), Both Set Origin register (BSO).
11. In all examples instructions to enable the "CAPS LOCK" key refer to the AED 767 only.

Draw a blue circle (Note: blue circle has no holes at intersection with the yellow because it was drawn last)	SEC4. DCL100.
Set color to red	SEC1.
Do Boundary Fill to the blue boundary. (CAP is inside, and boundary is a closed blue curve)	BFL4.

Example 2: Show the effect on a closed curve and a BFL to it,
when the curve is intersected by another curve.

TERMINAL DESCRIPTION -----	COMMANDS -----
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Move CAP to {250,250}	MOV250.250.
Draw yellow circle with a radius of 127	SEC3. DCL127.
Move CAP to {400,110}	MOV400.110.
Draw a blue circle with a radius of 110	SEC4. DCL110.
Return CAP to inside the yellow circle	MOV250.250.
Set current color to red	SEC1.
Do Boundary Fill to the yellow boundary	BFL3.
(Note - the red fill leaks beyond the yellow boundary, to fill the entire screen (except for the yellow curve)	

The difficulty, in filling to a boundary that has been overdrawn by another, may be removed by restoring the continuity/closure before doing the fill of the desired boundary. This is done by redrawing the entire boundary (in this example, the yellow circle) or by using the WPX, Write Pixel, command to individually restore the overdrawn pixels to the appropriate color (blue changed to yellow, in this example).

APPLICABLE FOR AED 767 ONLY

BLG*

```
-----  
set Blue Line Grid mode*  
-----  
FUNCTION CODES  
-----  
3 CHARS ASCII DECIMAL OCTAL HEX  
-----  
BLG      %      37      045    83  
-----
```

BLG*

PURPOSE:

The Blue Line Grid (BLG) command allows the user to turn the hardware blue line grid on and off.

*Extended Function (Must be preceded by a "+" in single character mode).

FORMAT:

```
-----  
BLG  switch  
-----
```

Where:

switch (0~1, o-type) indicates the status of the blue line grid and is defined as follows:

- 0 - the blue line grid is turned off (invisible)
- 1 - the blue line grid is turned on (visible)

USAGE:

The BLG command allows the user, under either program or keyboard control, to make the blue line grid visible or invisible.

NOTES:

- a. If the interpreter is in one character non-mnemonic encoding (see SEN command) then the user must prefix this command with a + (enter extended command mode). This sequence is not necessary in 3 character mnemonic encoding mode or if the host computer is capable of transmitting 8 bits of data. In the latter case the hex FUNCTION CODE should be used.

- b. The grid is non-destructive, that is it does not occupy video memory and therefore does not have any permanent effect on images displayed on the screen when it is turned on or off.
- c. The grid appears on the screen as a heavy blue line every 64 pixels, a medium blue line every 32 pixels, and a light blue line every 8 pixels.
- d. The grid intensity can be adjusted with the grid adjust pot located on the terminal back plane.

BSO

Both Set Origin registers				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
BSO	g	103	147	67

BSO

PURPOSE:

The Both Set Origin Register (BSO) command changes the position in video memory at which the upper left corner of the display window is located. This single command may be used to functionally replace the SHO and SVO commands.

FORMAT:

BSO | x10 y10

Where:

x10 (0 ~ 511/1023, c-type) is the x-coordinate of the pixel in video memory at which the raster display will originate;

y10 (0 ~ 511/1023, c-type) is the y-coordinate of the pixel in video memory at which the raster display will originate;

USAGE:

The BSO command allows the terminal programmer to simultaneously reset both the horizontal and vertical origin registers. These registers determine the location in video memory at which the monitor scan originates. Changing the contents of these registers has the effect of moving (panning) the display window across a fixed image in memory.

NOTES:

1. The default values are $x10 = 0$; $y10 = 482$ or 511 (512), 482 or 575 (767).

EXAMPLE:

Illustrate the use of BSD by generating the figure shown in Figure A, then moving display origins to produce Figures B and C.

DESCRIPTION -----	TERMINAL COMMANDS -----
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Move CAP to {255,255}	MOV255.255.
Draw fat circle of radius 5; do an interior fill in red (default)	DFC5. IFL
Set current color to green	SEC2.
Move CAP to {255,241}, the center of 512 x 483 window	MOV255.241.
Draw fat circle, radius 5	DFC5.
Write a pixel at circle center	WPX2.
Set current color to blue	SEC4.
Draw fat circle of radius 30 (Display is that of Fig. A)	DFC30.
Move display origin to center of red ball (Display is that of Fig. B)	BSO255.255.
Move display origin to center of green/blue circles (Display is that of Fig. C)	BSO255.241.

COP

COPY to AUX				

FUNCTION CODES				

3 CHARS	ASCII	DECIMAL	OCTAL	HEX

COP)	41	051	29

COP

PURPOSE:

The Copy (COP) command allows the user to copy the image in video memory onto a hardcopy device.

FORMAT:

```
-----  
COP ! device ncopies  
-----
```

Where;

device (0~15, o-type) is the device number of the hardcopy unit

ncopies (0~15, o-type) is the number of copies to be made

USAGE:

The COP command allows the user to utilize an ink on paper (color copy) device to obtain a hardcopy of the image presently in video memory.

DAI

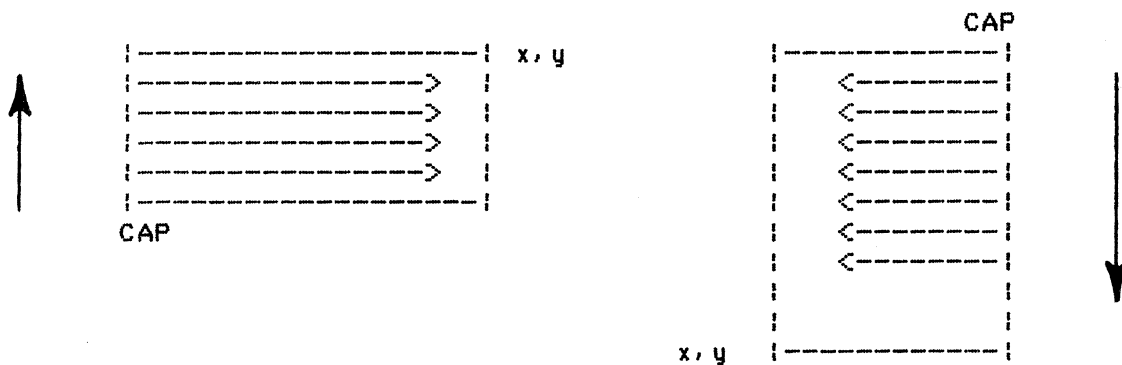
Define Area of Interest				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
DAI	r	114	162	72

DAI

PURPOSE:

The Define Area of Interest (DAI) command specifies a rectangular area to which the video memory scan may be restricted for purposes of reading or writing with subsequent commands (e.g. WHS,RHS,WHR,RHR, etc.). This area is referred to as the Area of Interest (AOI) and is defined as a rectangular area with one corner at CAP and the diagonally opposite corners at (x,y).

VIDEO MEMORY



FORMAT:

```
-----
DAI | X Y
-----
```

Where:

X (0 ~ 511/1023, c-type) is the x coordinate of the corner of the AOI opposite the corner at the CAP.

Y (0 ~ 511/1023, c-type) is the y coordinate of the corner of the AOI opposite the corner at the CAP.

NOTES:

1. If CAP x is less than the x coordinate specified in the DAI command then the scan will be left to right, otherwise the scan will be right to left. If CAP y is less than the y coordinate specified in the DAI command then the scan will be from bottom to top, otherwise the scan will be from top to bottom.
2. DAI applies only to run-length encoded read/write operations and DMA transfers to and from the AOI.

DCA

Draw Cursor Absolute

FUNCTION CODES

3 CHARS ASCII DECIMAL OCTAL HEX

DCA p 112 160 70

DCA

PURPOSE:

The Draw Cursor Absolute (DCA) command erases the cursor from the CAP, if necessary, and redraws it centered at a specified location.

FORMAT:

DCA : X Y

Where:

X (0 ~ 512/1023, c-type) is the x coordinate where the cursor is to be redrawn

Y (0 ~ 512/1023, c-type) is the y coordinate where the cursor is to be redrawn

USAGE:

The DCA command is a convenient way to visually indicate a position by drawing a cursor at that point. It may also be used to leave a marker (cursor) at the last position of the joystick after the DJC command is used. Once issued, the cursor will remain on the screen until the ECU command is issued. The DCA is overridden if the joystick is enabled. The cursor will remain when the joystick is disabled until an ECU is issued.

DCL

Draw Circle				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
DCL	0	79	117	4F

DCL

PURPOSE:

The Draw Circle (DCL) command will draw a circle in the current color with the radius specified and its center at the CAP.

FORMAT:

```
-----  
DCL | radius  
-----
```

Where:

radius (0~ 127, 0-type) is the radius of the circle measured in pixels.

USAGE:

The DCL command draws a circle with its center at CAP and a radius specified in the command.

NOTES:

1. Two concentric circles of radii differing by one pixel may have sporadic background color pixels visible between them. To avoid this when coloring and/or shading rings and spheres, use the Draw Fat Circle command (see DFC).

EXAMPLE:

<u>DESCRIPTION</u>	<u>TERMINAL COMMANDS</u>
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Move CAP to {100,200}	MOV100.200.
Draw a circle with a radius of 20 pixels in the current color	DCL20.
Set color to green (color 2)	SEC2.
Draw concentric green circle with a radius of 49 pixels	DCL49.

DFC

Draw Fat Circle				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
DFC	n	110	156	6E

DFC

PURPOSE:

The Draw Fat Circle (DFC) command causes a circle to be drawn so that concentric circles with a difference in radii of 1 may be drawn without any background pixels showing between the circles.

FORMAT:

```
-----  
DFC | radius  
-----
```

Where;

radius (0 ~ 127, O-type) is the circle radius measured in pixels.

USAGE:

When drawing a group of concentric shaded colored circles to produce what appears to be a colored sphere, in addition to the varying the hue of the color to provide shading, it is essential that none of the background pixels appear in the sphere interior. The DFC command allows drawing completely contiguous concentric circles.

EXAMPLE:

<u>DESCRIPTION</u>	<u>TERMINAL COMMANDS</u>
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Move CAP to {127,127}	MOV127.127.
Draw circles of radius 63, 62, 61 and 60 pixels respectively	DCL63. DCL62. DCL61. DCL60.
Move CAP to {255,127}	MOV255.127.
Draw fat circles of radius 63, 62, 61, and 60, pixels respectively	DFC63. DFC62. DFC61. DFC60.
Zoom display X2 after setting display origins	BS063. 240. SZR2. 2.

Notice background pixels showing through in the left circular band, and the absence of background pixels in the circles to the right where the DFC commands were used.

APPLICABLE FOR AED 767 ONLY

Draw Filled Polygon

DFP

FUNCTION CODES

DFP

3 CHARS ASCII DECIMAL OCTAL HEX

DFP ! 33 041 21

PURPOSE:

The Draw Filled Polygon (DFP) command allows the user to draw a polygon and fill it by supplying a number of xy coordinates. Used only by the AED 767. The DFP requires the use of one temporarily RESERVED working color which may be selected by the user. This color must not exist in the background of the polygon. The working color used is employed by the algorithm during the construction at the complex polygon but completely disappears from the screen at the conclusion of the command.

FORMAT:

DFP wc n P1...Pn

Where;

wc (0 ~ 255), o-type is a temporary working color

n (2 ~ 32767, o-type) is the number of xy coordinates which define the polygon ^{x/6}

p1...pn (0 ~ 1023, c-type) are the xy coordines of the polygon

USAGE:

The DFP (draw filled polygon) command allows the user to draw a polygon of any shape, boundary filled with the current color. The user need not specify the starting xy coordinate again as the last coordinate since the AED 767 will automatically close the polygon and fill it. Hence for a triangle the user need only supply three sets of coordinates.

NOTES:

- a. The polygon may cross over its own boundaries. For example, a figure "8" will be correctly filled.
- b. The DFP command uses solid fill only. If a stipple pattern has been selected, it will be ignored by this command.
- c. Beginning with Version 82.9, a flag color must also be designated. For example, if color location #255 is the flag color, then you would type

DFP255. 4. 100. 100. 300. 200. 300. 100. 100. 300.

The flag color must be different from the working color. The flag color outline will disappear with completion of the working color fill.

EXAMPLE:

Example 1:

DESCRIPTION -----	TERMINAL COMMANDS -----
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Set the current color to blue (4)	SEC4.
Draw "bow tie" polygon with four points and one "crossover"	DFP7. 4. 100. 100. 300. 200. 300. 100. 100. 300.

Example 2:

TERMINAL	
DESCRIPTION -----	COMMANDS -----
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Draw polygon with five points	DFP7. 5. 100. 100. 400. 100. 400. 500. 300. 300. 100. 300.

DFR

Draw Filled Rectangle

FUNCTION CODES

3 CHARS ASCII DECIMAL OCTAL HEX

DFR o 111 157 6F

DFR

PURPOSE:

The Draw Filled Rectangle (DFR) command is used to draw and fill a rectangle in the current color. If a stipple pattern has been previously set (see SPF), the rectangle will be filled with the selected pattern in the current color. One corner of the rectangle will be at the CAP, and the diagonally opposite corner will be at (x,y). The new CAP, after execution of this command, will be at (x,y). See FRR (Fill Rectangle Relative) command.

FORMAT:

DFR I X Y

Where:

X (0 ~ 511/1023, c-type) is the x coordinate of the corner of the rectangle opposite the CAP.

Y (0 ~ 511/1023, c-type) is the y coordinate of the corner of the rectangle opposite the CAP.

USAGE:

In addition to the use of DFR for drawing colored rectangles, other uses are possible. To erase a rectangular area, set the current color to the background color and draw a filled rectangle in the desired area. One rectangle may be drawn within another, and the inner rectangle colored to the background, producing a hollow rectangle as shown in the example below.

EXAMPLE:

Draw a green rectangle 200 pixels high by 512 wide at the bottom of the display. Centered within that rectangle draw another rectangle 100 pixels high by 412 pixels wide in white.

<u>DESCRIPTION</u>	<u>TERMINAL COMMANDS</u>
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Move the CAP to {50,50}	MOV50. 50.
Set the current color to green {2}	SEC2.
Draw filled rectangle to {462,250}	DFR462. 250.
Move CAP from {462,250} to {100,100}	MOV100. 100.
Set current color to white {7}.	SEC7.
Draw filled rectangle to {400,200}	DFR400. 200.

DJC

Disable Joystick Cursor

FUNCTION CODES

3 CHARS ASCII DECIMAL OCTAL HEX

DJC d 100 144 64

DJC

PURPOSE:

The Disable Joystick Cursor (DJC) command is used to disable the joystick cursor which was enabled by the EJC command.

FORMAT:

DJC

NOTES:

CAP remains at the last cursor location before the DJC command is issued. The DJC command will not erase a cursor placed on the screen by the DCA (Draw Cursor Absolute) command.

DMV

Draw Multiple Vectors				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
DMV	m	109	155	6D

DMV

PURPOSE:

The Draw Multiple Vectors (DMV) command is equivalent to multiple DVR commands. A single function code is followed by a string of vector end points, defined by relative movements (dx,dy) from the previous end-point or CAP for the first set of vector end points.

FORMAT:

```
-----  
DMV | dx1  dy1 dx2  dy2  ....  dxn  dyn  0  0  
-----
```

Where:

dx (-128 ~ +127, 0-type) defines the change from the previous vector endpoint or CAP in the x direction.

dy (-128 ~ +127, 0-type) defines the change from the previous vector endpoint or CAP in the y direction.

USAGE:

DMV avoids needless repetition of the DVR command where connected vectors of the same color are to be drawn. The list of endpoints is terminated by specifying both dx and dy as 0 (zero).

EXAMPLE:

Draw a triangle starting at {200,256}.

<u>DESCRIPTION</u>	<u>TERMINAL COMMANDS</u>
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Move CAP to {200,256}	MOV200.256.
Draw multiple vectors;	DMV50.127.50.-127.-100.0.0.0.

DPA

Disable PAnning				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
DPA	&	38	46	26

DPA

PURPOSE:

The Disable Panning (DPA) command is used to terminate the panning mode which is enabled by the Enable Panning (EPA) command.

FORMAT:

DPA

USAGE:

The DPA command will disable control of panning through the joystick and cause the image to remain in the last panned position before the DPA was issued.

DPK

Define Programmable Key				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
DPK	N	78	116	4E

DPK

PURPOSE:

The Define Programmable Key (DPK) command allows the user to transmit a sequence of up to 8 commonly used characters with a single keystroke. Also, see DSK (Define Soft Key) command.

FORMAT:

```
-----  
DPK : keyno [code1 code2 ... code8] 0  
-----
```

Where:

keyno (0 ~ 7 o-type) is the user key on the AED 512/767 keyboard whose function is to be redefined.

code (0 ~ 127 o-type) is the ASCII representation of the characters to be transmitted.

USAGE:

The DPK function allows often used keystroke sequences to be replaced (from the keyboard operators point of view) by a single keystroke. Please note, DPK can only be used by depressing the particular programmed key.

NOTES:

1. If the special function key has been redefined using the DPK command, then the normal (default) function of that key is no longer available from the keyboard.
2. Terminal reset from either the keyboard or by command (see RST) will reset all special function keys to their normal (default) power-up settings.

EXAMPLE:

Example 1:

For example, to make the Special Function Key number 1 (which normally controls the AED 767 anti-aliasing mode) produce the word PRINT followed by quotation marks when pressed, the following sequence would be used.

<u>DESCRIPTION</u>	<u>TERMINAL COMMANDS</u>
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Redefine user key 1 to transmit:	DPK1.
P	80.
R	82.
I	73.
N	78.
T	84.
Quotation marks (")	34.
Terminate the sequence	0.

If the terminal is in the LOCAL and TEXT mode, each time Special Function Key number 1 is depressed the 6-character group PRINT followed by a quotation marks will appear on the monitor.

If the terminal is in the Remote (Host Communications mode), PRINT followed by a carriage return will be sent to the host.

This command may also be used to generate graphics commands sequences. For example, to program a key to draw, in the current color, a circle at CAP of 65 pixels radius and perform an interior fill from the center of the circle; the following sequence would be used:

Example 2:

DESCRIPTION -----	TERMINAL COMMANDS -----
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN1888N
Redefine key 1 to transmit:	N^A
Draw a circle with a 65 pixel radius	OA
Do an interior fill	I
Terminate with 0	SHIFT^P

DRL

Define Register LED				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
DRL	=	61	075	3D

DRL

PURPOSE:

The Define Register LED (DRL) command causes the row of eight Light Emitting Diode (LEDs) indicator lights (LEDs), at the top-right of the keyboard, to indicate the contents of value. Figure 4.1 shows the location of the indicators (just above the user special function keys).

FORMAT:

```
-----  
DRL i value  
-----
```

Where:

value (0 ~ 255 o-type) is the single byte representation of the 8 user LED's to be turned on or off.

USAGE:

The most significant bit (MSB) of the byte corresponds to the leftmost LED in the group and so on down to the least significant bit (LSB) which corresponds to the rightmost LED in the group. If a bit is on (1) then the corresponding LED is turned on, if the bit is off (0) then the corresponding LED is turned off.

These indicator lights are used to give the terminal operator visual quantitative feedback. For example, the LED's might be set to indicate the current layer number being displayed in a multi-layer VLSI design.

NOTES:

1. The status of the LED (on or off) has no relationship to the status of the special function key which appears below it.

EXAMPLE:

<u>DESCRIPTION</u>	<u>TERMINAL COMMANDS</u>
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Turn on all 8 LED's	DRL255.
Turn on the even numbered LED's (0,2,4,6) only	DRL170.
Turn off all of the LED's	DRLO.

DSF

Define Special Font				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
DSF	7	55	067	37

DSF

PURPOSE:

The Define Special Font (DSF) command places a user specified pattern in the AED 512/767 RAM which may be used to produce a special purpose symbol on the monitor. Also, see WSF and ESF.

FORMAT:

```
-----  
DSF | code  hsize  vsize  [mask  byte1 byte2 ... byten] 0  
-----
```

Where:

- code (1 ~ 255, o-type) is a nonzero number used to identify the character (font).
- hsize (1 ~ 255, o-type) determine the horizontal and vertical dimensions and vsize of the matrix of pixels containing the character (font).
- vsize (1 ~ 255, o-type) determine the horizontal and vertical dimensions and vsize of the matrix of pixels containing the character (font).
- mask (0 ~ 255, o-type) is a byte defining which planes of video memory will be used to address the color table when the character is written with the WSF command.
- byte (0 ~ 255, o-type) The byte(s) whose bit patterns determine the pixel-by-pixel writing which forms the character (font) being generated.
- 0 Termination zero.

NOTES:

1. Regardless of matrix dimensions, the special font character is defined bit by bit in 8 bit byte groups starting with the bottom left most pixel and going from left to right, bottom to top, row by row. If a bit is on (1) then the corresponding pixel in the pattern will be on, if a bit is off (0) the background pixel will be untouched.
2. To calculate the number of byte(s) needed to represent a pattern; multiply the pattern's width by its height, divide by 8 (8 bits/byte), and round up to the nearest integer.

EXAMPLES:

Two special fonts shown below illustrate how the DSF command enables the user to generate two separate patterns.

TERMINAL DESCRIPTION -----	COMMANDS -----
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Define font number 1	DSF1.
to be a 8 pixel by 2 pixel pattern which will be written into memory plane 1 only	8. 2. 1.
and to have the bit pattern: 10001000 00110011	136. 33.
and terminate the command	0.
Move CAP to (255,255)	MOV255.255.
Display font twice	WSF1. 10. 10. 1. 0. 0. 0.

Define font number 2	DSF2.
to be a 3 pixel by 6 pixel pattern which will be written into memory plane 2 only	3.6.2.
and to have the bit	144.153.64.
pattern: 100 100 001 001 100 101	
and terminate the command	0.
Display font twice	WSF2.10.10.2.0.0.0.

DSK

```
-----  
Define Soft Key  
-----  
FUNCTION CODES  
-----  
3 CHARS ASCII DECIMAL OCTAL HEX  
-----  
DSK      %      37      45      25  
-----
```

DSK

PURPOSE:

The Define Soft Key (DSK) command is similar to the DPK command in its function, except that more bytes can be stored and then recalled by a single key depression.

FORMAT:

```
-----  
DSK | keyno  n  code1 code2 ... coden  
-----
```

Where:

keyno (0~7, o-type) is the user key number on the AED 512/767 to be redefined

n (1~15, o-type) is the number of bytes to be transmitted when the user key is depressed

code (0~255, o-type) is the ASCII representation of the character to be transmitted

USAGE:

The DSK function allows often used keystrokes or character sequences to be replaced (from the keyboard, operators point of view) by a single key stroke.

NOTES:

1. If the function special function key has been redefined using the DPK command, then the normal function of that key is no longer available from the keyboard.
2. Terminal reset from either the keyboard or by command (see RST) will reset all special function keys to their default power up settings.

EXAMPLE:

For example, to make Special Function Key No. 1 produce the sequence PRINT-AED:

<u>DESCRIPTION</u>	<u>TERMINAL COMMANDS</u>
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Redefine user key number 1	DSK1.
to transmit the following 9 bytes:	9.
PRINT	80.82.73.78.84.
-AED	45.65.69.68.

Reference to the ASCII chart, p. 4-8, will show the derivation of the numeric values (eg. 80 = P, 82 = R etc.)

If the terminal is in the LOCAL TEXT mode, each time Special Function Key No. 1 is depressed the 9-character group PRINT-AED will appear on the monitor. The DSK command may also be used to generate graphic commands when the AED 512/767 is in INTERPRETER and LOCAL mode.

?

AED 712 ONLY

DSP*

```

-----
Define Stipple Pattern*
-----
FUNCTION CODES
-----
3 CHARS ASCII DECIMAL OCTAL HEX
-----
DSP      !      33      041      7F
-----

```

DSP*

PURPOSE:

The Define Stipple Pattern (DSP) command allows the user to define a pattern only to be used in filling rectangles. After defining the pattern, the Set Pattern Fill (SPF) command must be issued.

*Extended function (must be preceded by a "+" in single character mode).

FORMAT:

```

-----
DSP npat pat1...pat8
-----

```

Where:

npat (0~15, o-type) is the number of the stipple fill pattern to be used with subsequent Draw Filled Rectangle (DFR) or Fill Rectangle Relative (FRR) commands.

pat1...pat 8 (0~255, o-type) define the pattern to be associated with pattern number npat.

USAGE:

The DSP command allows the user to define a pattern to be used when a filled rectangle is drawn with either the DFR (Draw Filled Rectangle) or the FRR (draw Filled Rectangle Relative) commands.

The pattern is defined on an 8 pixel by 8 pixel "checkerboard". This checkerboard pattern is repeated in the rectangle until the rectangle is completely filled. The rectangle is filled according to which diagonal corners are defined. Checkerboard pattern is defined as follows: Each square on the checkerboard corresponds to a single pixel on the screen. If that pixel or square is to be turned on (visible) in the pattern then the square is filled with a 1, otherwise the square is filled with a 0. Each row of the checkerboard is then converted to an 8-bit binary number (high order or bit 7 is the left-hand end of the row) and the numbers become the values pat1 through pat8.

NOTES:

- a. If the interpreter is in one character non-mnemonic encoding (see SEN command) then the user must prefix this command with a + (enter extended command mode). This sequence is not necessary in 3 character mnemonic encoding mode of if the host computer is capable of transmitting 8 bits of data. In the latter case the hex function code should be used.
- b. The pattern is always written in the current color (see SEC command and is "filtered" by any write masks currently active (see SWM command).
- c. Regardless of where the rectangle begins or ends, the stipple pattern is referenced to the (0,0) memory coordinate, so overlaid patterns always coincide regardless of their respective overlapping rectangle positions.

EXAMPLE:

The following example will show how to fill a rectangle with its lower lefthand corner (10, 10) and its upper righthand corner at (100, 100). The rectangle will be filled with a pattern of X's. Our "checkerboard" for such a pattern would appear as follows:

```

0 0 0 0 0 0 0 0
0 X 0 0 0 0 0 X 0
0 0 X 0 0 0 X 0 0
0 0 0 X X 0 0 0
0 0 0 X X 0 0 0
0 0 X 0 0 X 0 0
0 X 0 0 0 0 X 0
0 0 0 0 0 0 0 0

```

Assigning the proper binary values to each square, we would get the following numbers shown in the fifth of the checkerboard:

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0	66
0	0	1	0	0	1	0	0	36
0	0	0	1	1	0	0	0	24
0	0	0	1	1	0	0	0	24
0	0	1	0	0	1	0	0	36
0	1	0	0	0	0	1	0	66
0	0	0	0	0	0	0	0	0

The command to define the above pattern would be:

DESCRIPTION -----	TERMINAL COMMANDS -----
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Define stipple fill pattern no. 1	DSP1. 0. 66. 36. 24. 24. 36. 66. 0.
Enable stipple fill pattern no. 1	SPF1.
Positon CAP to {10, 10}	MOV10. 10.
Draw a stipple filled square 90 x 90 pixel	DFR100. 100.

DTM

Define Table Mapping				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
DTM	2	50	062	32

DTM

PURPOSE:

The Define Table Mapping (DTM) command establishes the scaling between horizontal and vertical dimensions on a graphics tablet surface and the corresponding dimensions in video memory.

FORMAT:

```
DTM ! xorigin yorigin xscale yscale
```

Where:

xorigin (0~65535, o-type) and yorigin are the tablet coordinates of the lower left-hand corner of the tablet rectangle which will be mapped into video memory, in units of thousandths of an inch.

xscale (0~255, o-type) and y scale are calculated from:
 $xscale = 65536/xsize$
 $yscale = 65536/ysize$

xsize (0~11000, o-type) and ysize are dimensions, in thousandths of an inch, of the tablet rectangle to be mapped into video memory.

Before scaling the tablet to video memory, align the tablet unmarked electronic axis and (0, 0) origin with the corresponding screen axis and (0, 0) origin, respectively. This is a trial-and-error process which can be devised by the operator; however, one method is to draw orthogonal vectors from (0, 0) on the screen or use the Blue Line Grid (767 only), then adjust a sheet of grid paper or transparent plastic on the tablet, so horizontal and vertical sweeps of the tablet stylus (pen) from its (0, 0) origin correspond to the axis system on the screen. When satisfactorily adjusted, affix the grid to the tablet. Now the origin and scaling corrections can be made in the command DTM.

NOTES:

1. The graphics tablet is always mapped to the entire 512/767 screen.

DVA

Draw Vector Absolute

FUNCTION CODES

3 CHARS ASCII DECIMAL OCTAL HEX

DVA A 65 101 41

DVA

PURPOSE:

The Draw Vector Absolute (DVA) command draws a vector, in the current color from CAP to the point {x,y}. The new CAP is at {x,y}. Also, see AAV, DVR, and SLS commands.

FORMAT:

DVA | X Y

Where:

X and Y (0~511/1023, c-type) are the coordinates of the vector endpoint.

USAGE:

This command is used to draw a straight line segment from the Current Access Position (CAP) to a specified point, anywhere in video memory. The specified point (x, y) is described in absolute coordinates rather than relative to the current position of the CAP. Dotted line patterns of single width lines can be drawn by invoking the SLS command before issuing the DVA command.

EXAMPLE:

<u>DESCRIPTION</u>	<u>TERMINAL COMMANDS</u>
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Move the CAP to {200,256}	MOV200.256.
Draw vector to {250,383}	DVA250.383.
Draw vector to {300,256}	DVA300.256.
Draw vector to {200,256}	DVA200.256.

DVR

----- Draw Vector Relative ----- FUNCTION CODES -----				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
DVR	1	108	154	6C

DVR

PURPOSE:

The Draw Vector Relative (DVR) command draws a vector from CAP to CAP+{dx,dy}. The new CAP is at this termination. See also, AAV, DVA, and SLS commands.

FORMAT:

```
-----  
DVR  dx  dy  
-----
```

Where:

dx (-128~127, o-type) is the change in x, in pixel units, of the vector endpoint

dy (-128~127, o-type) is the change in y, in pixel units, of the vector endpoint

USAGE:

The draw relative command allows the user to determine the endpoint of a vector relative to the CAP. Also, since the coordinates are o-type, it takes fewer bytes to describe a vector. This can be very important when host-terminal transmission is being done through serial communications. The default condition is to draw vectors with single width solid lines. Alternatively dotted line patterns of single width lines can be drawn by invoking the SLS command.

NOTES:

Vectors being drawn with the DVR command may not exceed 127 in length for either the x or y direction

EXAMPLE:

Draw the diagrammed triangle, using DVR commands.

EXAMPLE:

<u>DESCRIPTION</u>	<u>TERMINAL COMMANDS</u>
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Move the CAP to {200,256}	MOV200.256.
Draw vector AB	DVR50.127.
Draw vector BC	DVR50.-127.
Draw vector CA	DVR-100.0.

This is the same triangle drawn in the example of DVA command.

ECU

Erase Cursor Unconditionally				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
ECU	5	53	065	35

ECU

PURPOSE:

The Erase Cursor Unconditionally (ECU) command allows the user to erase the cursor from the screen.

FORMAT:

ECU

USAGE:

The ECU command is used to remove the cursor from the display after it has been placed there by using the DCA command.

NOTES:

1. DCA and ECU should be used as a pair, as should EJC and DJC. The use of EJC to turn the cursor on and then ECU to turn it off produces anomalous results and the cursor is not erased.

EXAMPLE:

Erase the cursor using the DCA, ECU pair.

DESCRIPTION -----	TERMINAL COMMANDS -----
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Move to screen center	MOV256.242.
Write pixel in white, Draw red circle	WPX7.DCL20.
Draw cursor at absolute location {256,242} to show center of circle	DCA256.242.
Erase cursor	ECU

EJC

----- Enable Joystick Cursor -----				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
EJC	U	85	125	55

EJC

PURPOSE:

The Enable Joystick Cursor (EJC) command allows the user to enable the joystick cursor.

FORMAT:

EJC

USAGE:

The EJC command turns on the cursor so it graphically displays the joystick position. Alternatively, the user can depress the CURSOR special function key. To extinguish the joystick cursor, invoke the DJC command or depress the CURSOR special function key. CAP will be the visible cursor position.

NOTES:

The user may depress the RATE or VERNIER terminal keys to enable/disable these functions. Alternately, these functions may be set/reset by the SCS (Set Console Status) command.

EXAMPLE:

Enable the joystick cursor.

DESCRIPTION -----	TERMINAL COMMANDS -----
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Move to {256,242}	MOV256.242.
Write pixel in white, draw red circle	WPX7.DCL20.
Enable joystick cursor control [Green X cursor appears on screen]	EJC
Manually move the joystick, and hence the cursor, to center the cursor on the white pixel. When cursor is near the desired point, depress "VERNIER" special function key to aid final positioning; depress again to disable VERNIER.	
Disable joystick cursor control.	DJC

APPLICABLE ON AED 767 ONLY

ELP*

```

-----
                draw ELLiPse
-----
                FUNCTION CODES
-----
3 CHARS ASCII DECIMAL OCTAL HEX
-----
ELP      "      34      042    80
-----

```

ELP*

PURPOSE:

The Draw Ellipse (ELP) command allows the user to draw an ellipse.

*Extended function (must be preceded by "+" in single character mode).

FORMAT:

```

-----
ELP  xsize ysize const
-----

```

Where:

x size (1~255, o-type) is the size of the ellipse on the horizontal axis

y size (1~255, o-type) is the size of the ellipse on the vertical axis

const (1~128, 0-type) is a constant calculated by the user

USAGE:

The ELP command allows the user to draw an ellipse anywhere on the screen. The ellipse's axis must be either parallel to the X axis or Y axis. Because of machine restrictions of the processor used in the AED 767 the user must calculate a constant for the ellipse command. The calculation of the constant is defined as follows:

$$F = xsize/ysize$$

If F is greater than 1 then $F = 1/F$

$$const = 255 * F^2 \quad \text{rounded to the nearest integer}$$

Note: In the above, standard FORTRAN notation has been used (i.e. * means multiplication, / means division, and ** means exponentiation).

NOTES:

- a. If the interpreter is in one character non-mnemonic encoding (see SEN command) then the user must prefix this command with a + (enter extended command mode). This sequence is not necessary in 3 character mnemonic encoding mode or if the host computer is capable of transmitting 8 bits of data. In the latter case the hex FUNCTION CODE should be used.
- b. The ellipse is drawn in the current color and in accordance with the current write mask (see SWM command).

EXAMPLE:

DESCRIPTION -----	TERMINAL COMMANDS -----
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Activate Cursor	EJC
Move cursor to center of screen	
Formula: If x size > y size then 255 (y size/x size) or else 255 (x size/y size). Round to nearest interger. Therefore to draw an ellipse 60 pixels B wide x 30 pixels high: $(30/60) = 1/4; 255 \times 1/4 = 64$	
Draw ellipse along the x axis	ELP60.30.64.
Draw ellipse along the y axis 50 pixels wide x 100 pixels high	ELP50.100.64.

EPA

Enable Panning				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
EPA	h	104	150	68

EPA

PURPOSE:

The Enable Panning (EPA) command allows the user to pan the visible display window vertically and horizontally or both in order to display a selected portion of video memory.

FORMAT:

EPA

USAGE:

Often the terminal video memory contents extend beyond the "window" of visibility provided by the terminal display. The EPA command is used to "pan" an image via the joystick.

NOTES:

The EPA command will perform the identical function as hitting the pan button once. Joystick x and y information will continually update origin registers without host intervention.

EXAMPLE:

Draw a 100 pixel radius circle and, with EPA, pan it around the screen.

DESCRIPTION -----	TERMINAL COMMANDS -----
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Erase screen	ERS
Change CAP to {255,255}	*MOV255.255.
Draw circle	DCL100.
Enable panning	EPA
[Move joystick to move (pan) the circle.]	
Disable pan	DPA [Note that moving the joystick no longer pans the circle.]

*Note: For 767, substitute MOV 512.512 for, MOV 255.255.

ERS

ERaSe entire memory				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
ERS	~	126	176	7E

ERS

PURPOSE:

The Erase Entire Memory (ERS) command erases the entire video memory write-enabled planes only. ERS should be issued after returning zoom to normal viewing.

FORMAT:

ERS

USAGE:

ERS (Erase...) compares with FFD (Form Feed) as shown.

ERS	FFD
-----	-----
Origins unchanged	Origins restored to default
Does not exit the interpreter	Exits the interpreter
Once panned screen wraps around	After FFD issued screen will scroll again
- Erases video memory, write-enabled planes only -	

NOTES:

1. Also see also the Set Write Mask command (SWM).

ESC

ESCAPE				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
ESC	ESC	27	033	1B

ESC

PURPOSE:

The Escape (ESC) command causes the terminal to change from the Alphanumeric Mode to the graphics Interpreter Mode. Also see XXX command.

FORMAT:

ESC (or) CTRL [

USAGE:

After "power-up" or "terminal reset", the terminal will be in Alpha (Text) Mode. The ESC command will typically be used to change the state of the terminal into the Interpreter Mode.

NOTES:

*Different results from same command in hex. Difference in commands is apparent in 3-character encoding.

EXAMPLE:

<u>DESCRIPTION</u>	<u>TERMINAL COMMANDS</u>
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Leave interpreter	XXX (eXit, graphiX, enter teXt)
To get "A" displayed	A
To enter interpreter mode	ESC Key
Move CAP to {100, 250}	MOV100.250.
Draw vector to {0, 0}	DVA0.0.
Change from INTERPRETER to TEXT mode	XXX
Character sequence to get "A00" to appear on display	A00

ESF

Erase Special Font				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
ESF	9	57	071	39

ESF

PURPOSE:

The Erase Special Font (ESF) command erases, by changing to the background color, an area hsize by vsize with its lower left-hand corner at the CAP; the command then updates CAP to CAP+(dx,dy). See also the WSF command.

FORMAT:

```
-----  
ESF | hsize | vsize | dx | dy  
-----
```

Where:

hsize are the horizontal and vertical dimensions and

vsize (0 ~ 255, 0-type) of the matrix containing the character to be erased

dx,dy (-128 ~ 127, 0-type) are coordinates of the incremental move of the CAP after the erasure

USAGE:

The ESF command performs a function similar to that of the combination SEC (Set Current Color) and DFR (Draw Filled Rectangle). However, with ESF, the color change is always to the background color for erasure, then automatically back to the previous current color.

ETC

----- Enable Tablet Cursor -----				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
ETC	3	51	063	33

ETC

PURPOSE:

The Enable Tablet Cursor (ETC) command enables/disables an auxiliary graphics digitizing tablet and sets overall readback format in conjunction with the STP command.

FORMAT:

```
-----
ETC idchar
-----
```

Where;

(idchar) is not encoded as an operand, but is exactly one character (0 ~ 255 , 8 bits)

idchar = 0 (ASCII NUL or CNTL-SHFT-P) Tablet disabled.

idchar = 1 ~ 255 (a single byte or character)

Tablet enabled. Cursor on terminal will return the sequence;

```
-----
idchar status x16 y16
-----
```

each time the stylus is depressed or a tablet cursor button is pushed. If two or more cursor buttons are simultaneously pushed and then one is released, the sequence will again be sent.

status = is a code corresponding to the button(s) pushed.

USAGE:

The ETC command is used to enable or disable an auxiliary graphics digitizing tablet normally connected to the RS-232C auxiliary serial port built into the AED 512/767.

NOTES:

1. The ETC command should be preceded by Define Table Mapping (DTM) which scales from the tablet area to the video memory.
2. If the pen or crosshair is outside the rectangular area of the tablet to be drawn upon (and which is precisely defined in DTM), 16 is added to the status code word. In this case the terminal returns the actual absolute tablet coordinates rather than video memory coordinates. This is useful for applications which use a rectangular digitizing "window" surrounded with menu select areas.
3. The ETC command is normally used with the STP (Set Tablet Parameters) command. STP controls conditions under which a "pen hit" and stylus coordinates are returned to a user program.

ETP*

----- Enable Tablet Polling -----				
FUNCTION CODES				

3 CHARS	ASCII	DECIMAL	OCTAL	HEX

ETP	+'	39	47	85

ETP*

PURPOSE:

The Enable Tablet Polling (ETP) command enables/disables a special polling mode of the AED 512/767 graphics tablet firmware. When this mode is enabled, stylus coordinates are sent only when requested with the RTP command, instead of upon pen down and/or pen up.

*Extended function (must be preceded by a "+" in single character mode).

FORMAT:

ETP | arg

Where:

arg is not encoded as an operand, but is exactly one character (0 ~ 255, 8 bits).

arg = 0 (ASCII NUL or CNTL-SHFT-P) Normal ETC functions

arg = 1 ~ 255 idchar, pen status coordinates are sent only on RTP command. Sequence is:

idchar STATUS x16 y16

FFD

Form Feed

FUNCTION CODES

3 CHARS	ASCII	DECIMAL	OCTAL	HEX
FFD	^ L	12	014	0C

FFD

PURPOSE:

The Form Feed (FFD) command does the following:

- a. Erases entire video memory, write-enabled planes only.
- b. Restores display origins to default power-on reset values.
- c. Causes exit from interpreter to alpha mode.
- d. Does not change current zoom levels (x and y).

FORMAT:

FFD

USAGE:

FFD (Form Feed) compares with the ERS (Erase Entire Memory) as shown.

FFD	ERS
----- Origins restored to default	----- Origins unchanged
Exits the interpreter	Does not exit the interpreter
	--Erases video memory, write-enabled planes only--
	-----No change in x,y zoom levels-----

NOTES:

- 1. See also the SWM (Set Write Mask) command.

Does not work ??

FRR

Filled Rectangle Relative				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
FRR	,	44	054	2C

FRR

PURPOSE:

The Filled Rectangle Relative (FRR) command is used to draw and fill a rectangle in the previously specified current color. If the SPF command is in effect, rectangle will be drawn in current color using the selected stipple pattern. FRR differs from DFR only in that arguments specified to FRR are displacements relative to the current CAP, not absolute coordinates.

FORMAT:

```
-----  
FRR i dx i dy  
-----
```

Where:

(x,y) is the location in video memory of the CAP and one corner of the rectangle (x+dx,y+dy) is the location of the diagonally opposite corner. It is also the new CAP.

dx,dy (-128 ~ +127, o-type) are the x and y displacements defining the horizontal and vertical sides of the rectangle to be drawn. Negative values for either coordinate are acceptable and will change the orientation of the rectangle and the CAP appropriately.

USAGE:

In many situations it is more convenient to specify the diagonally opposite corner of the rectangle by a displacement relative to the CAP rather than relative to the origin of coordinates.

GFL

General Fill				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
GFL	\$	36	044	24

GFL

PURPOSE:

The Generalized Overlay Fill (GFL) command fills the region surrounded by CAP with the current color until pixel boundaries of bcolor are encountered. bcolor = (current color) AND (mask).

FORMAT:

```
-----  
GFL : mask : bcolor  
-----
```

Where:

mask is a byte [M7, M6, ..., M1, M0] and:

[Mx]= 0: Nth memory plane disabled for boundary search
1: Nth memory plane enabled for boundary search

bcolor is the boundary color (0 ~ 255, 0-type)

USAGE:

The GFL command is primarily used to fill through various pixel groups until encountering a specific boundary type. This differs from the IFL command which fills until encountering any boundary.

EXAMPLE:

Draw three filled rectangles, red and green and blue, as in. Use all of the memory planes 0 through 6 (ie. seven video memory planes), but reserve plane 7 for overlay (GFL) use. Non-destructively overlay a white circle and fill it with green.

DESCRIPTION -----	TERMINAL COMMANDS -----
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Define color 128 to be white	SCT128. 1. 255. 255. 255.
Move CAP to lower left of display	MOV0. 0.
Draw and fill rectangle in red (default)	DFR200. 482.
Draw and fill rectangle in green	SEC2. DFR312. 0.
Set color to blue for and draw filled rectangle	SEC4. DFR511. 482.
Draw white circle so it overlays all three rectangles	MOV256. 241. SEC128. DCL127.
Set color to yellow and draw circle	SWM128.
Define color 136 = 128+8=100100 to be green	SCT136. 1. 0. 255. 0.
Set current color to 136	SEC136.
Do generalized fill, looking at memory through mask=128 and filling to boundary of bcolor=128.	GFL128. 128.

GS

```
-----  
Graphics System tektronix mode  
-----  
FUNCTION CODES  
-----  
3 CHARS ASCII DECIMAL OCTAL HEX  
-----  
GS      ^J      29      035    1D  
-----
```

GS

PURPOSE:

The Graphics System Tektronix Mode (GS) command draws vectors (and in the limit, single pixels) using the Tektronix emulation capability. After power-up, the AED terminals are immediately initialized and enabled to receive and execute Tektronix 4010 thru 4014 commands.

FORMAT:

```
-----  
GS | new.cap | (vector) | cc  
-----
```

Where:

new cap is the location in video memory to which the CAP will move before drawing the sequence of vectors 0 ~ 51 or on 1023 for superoam or 767, see below for encoding)

vector is the sequence (vector 1, vector 2, ..., vector n) of end points (0 ~ 51 or 1023, see below)

cc is a terminating control character, one of the following: CR(13), ESC(27), FS(28), RS(30) or US(31).

ENCODING OF COORDINATES

(Tektronix compatible)

Coordinates from host are sent in sequences from one to four characters:

AED 512 NORMAL

	Bit	7	6	5	4	3	2	1	0
1st Character	hi. y	XX	0	1	Y8	Y7	Y6	Y5	Y4
2nd Character	lo. y	XX	1	1	Y3	Y2	Y1	Y0	XX
3rd Character	hi. x	XX	0	1	X8	X7	X6	X5	X4
4th Character	lo. x	XX	1	0	X3	X2	X1	X0	XX

AED 512 SUPERDAM or AED 767

	Bit	7	6	5	4	3	2	1	0
1st Character	hi. y	XX	0	1	Y9	Y8	Y7	Y6	Y5
2nd Character	lo. y	XX	1	1	Y4	Y3	Y2	Y1	Y0
3rd Character	hi. x	XX	0	1	X9	X8	X7	X6	X5
4th Character	lo. x	XX	1	0	X4	X3	X2	X1	X0

NOTES:

1. Option switch 4 should be up to give command priority to 4014 emulation. This prevents conflicts between 4014 and normal 512/767 command interpretation.
2. Bit positions 5 and 6 in the coordinate specification determine 3 unique codes. Note: That hi. x and hi. y have the same code; this affects you when you try to abbreviate the 4-character sequence, explained next.
3. The four-character sequence may be abbreviated, omitting coordinates which do not change. Lo. x must always be sent as the last character of the sequence. If only lo. x changes, only lo. x need be sent. If lo. y or hi. y changes, just lo. y or hi. y followed by the unchanged lo. x need be sent. However, if only hi. x changes, the old lo. y must be sent first, then the new hi. x, then the old lo. x. The old lo. y sent first allows the Tektronix protocol to distinguish between hi. x and hi. y, since their codes in bits 6 and 5 are identical.

4. The Terminal Command Protocol has been structured such that the AED 512/767 is fully upward compatible with the non-raster Tektronix 4010 - 4014 series terminals. It is possible to run unmodified Plot-10 software to produce an appropriate display on the AED 512/767 consisting of vector graphics with alphanumeric legends. Graphics Input (GIN) mode will also be correctly emulated to allow the operator to use the AED 512/767's joystick and cursor without software modification. Additional commands to control unique terminal functions (e.g. color definition), can be integrated into most implementations without modification to existing host graphics software. The AED 767 has the added feature of Anti-Aliased Vectors (AAV) which smooths up to 8 colors of raster drawn vectors.
5. The AED 512/767 will perform complete 4010/4014 emulation (with or without the Enhanced Graphic Option, Option 34), including multiple line styles, incremental plot, random point plotting and multiple character sizes.
6. For maximum flexibility, Tektronix emulation modes can be entered from either alphanumeric mode or from the graphics interpreter.
7. Tektronix Special Point Plot Mode (random pixels with variable intensity) is emulated, but all points are the same intensity, and in the (previously set) current color. The random pixels are drawn with the repeating sequence (GS new.cap vector), where new cap = vector.
8. The AED 767 has two available window sizes for Tektronix emulation (512 X 512 on 1024 X 1024). This can be controlled by the STW command.

HOM

----- HOMe -----				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
HOM	_	95	137	5F

HOM

PURPOSE:

The Home (HOM) command returns the CAP, and hence the alpha cursor, to the first character position of the top line (that position depending on the current font size).

FORMAT:

HOM

USAGE:

This command is particularly useful when the alpha (underline) and graphics cursors are unlinked. (See example)

EXAMPLE:

Use the HOM command in the process of adding alpha titles to a graphic (circle), both with and without linking the CAP to the graphics cursor and joystick.

DESCRIPTION	TERMINAL COMMANDS
-----	-----
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Set Alpha Parameters: Double Size; 7 x 12 characters, unlinked graphics and alpha cursors.	SAP2746.60.U Note: No delimiters are used following 2, 7.

Move CAP to (256,200)	MOV256.200.
Draw circle of radius = 127 pixels	DCL127.
Send CAP home then exit the Interpreter Mode	HOM XXX
Type the title: "HOM-EXAMPLE"	HOM-EXAMPLE
Press return and line feed keys	RETURN LF
Push CURSOR function key, use joystick to move the cursor to the center of the circle.	
Type the word CIRCLE	CIRCLE
Note: That word CIRCLE is not displayed starting at green cursor, ie. CAP is not linked to the graphics cursor.	
Push cursor key, light goes out.	
Depress ESC key to change from Alpha to Interpreter Mode	ESC
Move CAP to (180,50)	MOV180.50.
Set Alpha Parameters to double the 5 x 8 size characters, and the CAP linked to graphics cursor/ joystick.	SAP2520.30.L
Leave the Interpreter Mode Type the letters CIR [Note where letters are displayed]	XXX
Push cursor key; use joystick to move the green X cursor to near mid-circle.	CIR

Type the letters CLE. CLE
[Note that the letters
are displayed atop one
another, at the cursor
location]

Push cursor key again
light goes out. [Green
cursor disappears, but
CAP is left where cursor
was]

Depress space bar, type CIRCLE
circle. [Now CAP advances
across the line]

Erase the screen; stay in
Interpreter Mode ERS

HSR

Horizontal Scroll Relative				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
HSR	w	119	167	77

HSR

PURPOSE:

The Horizontal Scroll Relative (HSR) command allows the user to set the horizontal origin register relative to the current horizontal origin register contents.

FORMAT:

```
-----  
HSR dx1  
-----
```

Where;

dx is a signed number.

USAGE:

The HSR command is used to set the horizontal origin with respect to its present setting [New = OLD + dx].

EXAMPLE:

Draw a 100 pixel radius circle and with the HSR command, divide the circle into two equal halves on the upper and lower portions of the display.

<u>DESCRIPTION</u>	<u>TERMINAL COMMANDS</u>
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
ENCODING	SEN3DDDN
Erase Screen	ERS
Move to {100,100}	MOV 100.100.
Draw a circle radius = 100	DCL 100.
Reset Horizontal Origin	HSR -100.

IFL

Interior Fill				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
IFL	I	73	111	49

IFL

PURPOSE:

The Interior Fill (IFL) command fills any complex closed polygon with the current color. Any pixel color other than the pixel color at CAP will serve as boundary pixels. CAP must be placed within the polygon to be filled.

FORMAT:

```
-----  
IFL  
-----
```

USAGE:

IFL colors (or floods) inside a previously drawn outline, starting at the enclosed CAP and moving to the nearest boundary (portion of a closed curve) in all directions. Closed curves which do not include the CAP (islands) are not colored.

NOTES:

1. The CAP is not guaranteed to remain in place through the IFL or other fill operations. A MOV command, to follow the fill, may be desirable.

EXAMPLE:

DESCRIPTION -----	TERMINAL COMMANDS -----
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Move CAP to {150,350}	MOV150.350.
Draw red circle of radius=50 and fill.	DCL50.IFL
Move CAP outside the circle; set color to white {7}	MOV300.300. SEC7.
Fill screen outside the circle.	IFL
[Fill is to video memory boundary]	
Erase screen to black	ERS
Move CAP to {250,250}	MOV250.250.
Draw yellow circle of radius=127	SEC3. DCL127.
Draw green circle, radius=80	SEC2. DCL80.
[Green should be an island within the yellow.]	
Move CAP to {400,110}	MOV400.110.
Draw blue circle, radius=110	SEC4.DCL110.
Set current color to 7	SEC7.
Move CAP inside green circle, Do interior fill [The "island" is now white]	MOV250.250. IFL

JUS

----- Jump to User Subroutine -----				
FUNCTION CODES				

3 CHARS	ASCII	DECIMAL	OCTAL	HEX

JUS	,	59	073	3B

JUS

PURPOSE:

The Jump to User Subroutine (JUS) command executes the user's subroutine beginning at the specified address in the AED 512/767 RAM.

FORMAT:

JUS ; addr

Where:

addr (0~65535, o-type) is the 16-bit address to be within the AED 512/767 to which control is to be transferred

USAGE:

The JUS command will cause control to be transferred to the address specified in the command. The executable code must have been previously loaded into the AED 512/767 by the user through the use of the LMR and SSE commands. It is necessary to know the firmware command entry points (addresses) to properly use JUS.

APPLICABLE ON AED 767 ONLY

LAT*

Load Anti-alias color Table*				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
LAT	&	38	046	84

LAT*

PURPOSE:

The Load Anti-aliasing Table (LAT) command allows the user to define a non-standard anti-aliasing color ramp.

*Extended function (Must be preceded by a "+" in single character mode).

FORMAT:

```
-----  
LAT  ibase  pval1...pval16  
-----
```

Where:

ibase (0~7, o-type) is the color table position in the first 8 color table position for which the ramp is being defined

pval1...pval16 (0 to 255) are the 16 ramps values (color table #5) they must conform to the methods for building anti-aliasing ramps described in the AAV command

USAGE:

The LAT command is invoked when the user wishes to anti-alias a color other than the first 8 default color or the user wishes to anti-alias to a background other than black.

NOTES:

- a. If the interpreter is in one character non-mnemonic encoding (see SEN command) then the user must prefix this command with a + (enter extended command mode). This sequence is not necessary in 3 character mnemonic encoding mode or if the host computer is capable of transmitting 8 bits of data. In the latter case the hex FUNCTION CODE should be used.
- b. Only 8 colors can be anti-aliased and they must reference color table positions 0 - 7. They must be laid out as described in the AAV command.

EXAMPLE:

Default color ramp for color 1 is:

129, 137, 145, 153, 161, 169, 177, 185, 193, 201, 209, 217, 225, 233, 241, 249

To change the ramp numbers enter the following:

LAT1. 130. 138. 146. 154. 162. 170. 178. 186. 194. 202. 210. 218. 226. 234. 242. 250.

AAV 2.
SEC 1.
MOV 0. 300.
DVA 500. 400.

You will see an anti-aliased green line with a red pixel on the end

Note: Color 1 is red but this drew a green line.

LMR

Load Microprocessor Ram				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
LMR	:	58	072	3A

LMR

PURPOSE:

The Load Microprocessor Ram (LMR) command loads n bytes of sub-routine (written in 6502 machine code) starting at a specified address in RAM, for later execution by use of the JUS command.

FORMAT:

```
-----  
LMR : addr n byte1 ... byten  
-----
```

Where:

addr (0~HI-RAM, o-type) is the address above the stack end of RAM, in 16-bit (2-byte) format (Refer to Address Map for free ram (HI-RAM) location)

n (0~65535, o-type) is the number of bytes of code to be downloaded

byte (0~255, o-type) byte/1 byte/2 byte/3...byten is the machine code subroutine, previously assembled.

USAGE:

1. The LMR command can be used, along with the SSE command, to allocate the RAM space between the top of addressable RAM and the dynamic storage area used by the interpreter. The JUS (Jump User Subroutine) can then be used to transfer from a host program to the program stored in the AED 512/767.
2. Depending on the amount of RAM storage space required for the subroutine, n bytes, the remaining RAM space is used as a LIFO (Last In, First Out) stack. The stored subroutine is protected from over-writing by proper choice of the address argument of the SSE (Set Stack End) command. If the stack (including storage requirements of DSF {Define Special Font} and {Define Special Key}) gets too large the SSE address is reached. Then the microprocessor exits whatever routine it is executing and returns the pointer to its value prior to the execution of the command that caused the memory overflow to occur.

APPLICABLE ON AED 767 ONLY

MAR*

Margin Controls*				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
LMR	:	58	072	3A

MAR*

PURPOSE:

The Margin Control (MAR) command is used to set the left and right text margins.

*Extended function (Must be preceded by a "+" in single character mode).

FORMAT:

```
-----  
MAR | left margin   right margin  
-----
```

Where: 0 < left/right margin < 1024
 left/right margin is c-type

USAGE:

1. Text margins can be set to the desired locations.
2. 767 users can write text beyond the 767 window by changing the margins.

NOTES:

1. MAR (0,0) resets the margins back to the default parameters (1,767).

MOV

----- MOVE absolute -----				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
MOV	Q	81	121	51

MOV

PURPOSE:

The Move Absolute (MOV) command sets the CAP (Current Access Position) to the location {x,y} in video memory.

FORMAT:

```
-----  
MOV ! x y  
-----
```

Where:

x (0~511/1023, c-type) is the x-coordinate.

y (0~511/1023, c-type) is the y-coordinate. See Note 1.

USAGE:

The MOV command is used to establish a particular location within the graphics field (CAP). This can be used to define the starting point of a vector or the center of a circle (before being drawn) etc.

NOTES:

1. If SUPERDAM (only for AED 512) has been invoked by use of the SUP1 command, the range of x and y arguments is extended to 0~1023.
2. The relationship between where the CAP is located in video memory (to commence drawing a vector, for example) and where the corresponding point will be written on the display depends on whether the display origin is displaced from its default position.

EXAMPLE:

Example 1: Paint a single pixel at display location {200,250} and in a default color.

DESCRIPTION -----	TERMINAL COMMANDS -----
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Move CAP to {200,250}	MOV200.250.
Write pixel in color 1	WPX1.

Example 2: This is valid only for the AED 512. Show the result of "off display" location commands in this way; paint a single pixel at {612,250} and in the default color.

AED 512 ONLY

DESCRIPTION -----	TERMINAL COMMANDS -----
INITIALIZE TERMINAL	RESET RESET LOCAL ESC
SET ENCODING	SEN3DDDN
Move CAP to {612,250}	MOV612.250.
Write pixel in color 1 (red)	WPX1.
Note: That the red spot (pixel) is painted at $x \text{ modulo } 512 = 100$, ie. wrapped around.	
Enable SUPEROAM	SUP1.
Move CAP TO {612,250}	MOV612.250.

(Example continued on next page)

Write pixel in color
2 (green)

WPX2.

Note: Green spot (pixel)
is not displayed
because viewing
window is over the
bottom-left quad-
rant of the SUPERDAM -
1024 x 1024 memory
and the pixel is in
the bottom-right
quadrant.

Display window to view
bottom-right quadrant

Note: Green pixel now
viewable at the
same relative
window position
as the red pixel
above.

SH0511.

MVR

----- MoVe Relative -----				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
MVR	i	105	151	69

MVR

PURPOSE:

The Move Relative (MVR) command sets the CAP to a new position {x+dx,y+dy} relative to the current access Position {x,y} just before execution. The new CAP is now at {x+dx,y+dy}.

FORMAT:

```
-----  
MVR i dx dy  
-----
```

Where:

dx (-128~127, o-type) is the change in x-coordinate

dy (-128~127, o-type) is the change in y-coordinate

USAGE:

The MVR command is similar in function to MOV but allows the x and y dimensions of a desired CAP move to be measured from the present CAP rather than from the origin of the coordinate system.

NOTES:

1. If SUPERDAM (only for AED 512) has been invoked by use of the SUP 1 command, the range of x and y arguments are extended to 0~1023.
2. The relationship between where the CAP is located in video memory (to commence drawing a vector, for example) and where the corresponding point will be written on the display depends on whether the display origin is displaced from its default position.

Draw a red spot at the center and corners of a square; put the center at (255,242) and make the square measure 100 pixels on a side.

DESCRIPTION -----	TERMINAL COMMANDS -----
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Move (absolute) the CAP to (255,242), mid-screen	MOV255.242.
Write red pixel at CAP (center of square)	WPX1.
Move CAP right 50 and up 50 pixels	MVR50.50.
Write red pixel at upper right corner	WPX1.
Move CAP to lower right corner	MVRO.-100.
Write red pixel at lower right corner	WPX1.
Move CAP to lower left corner	MVR-100.0.
Write red pixel at lower left corner	WPX1.
Move CAP to upper left corner	MVRO.100.
Write red pixel at upper left corner	WPX1.

OFL

Overlay Fill				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
OFL	V	86	126	56

OFL

PURPOSE:

The Overlay Fill (OFL) command fills a region surrounding the CAP bounded by pixels which are the same color as the current color. The resultant color of each pixel in the filled region will be determined by the logical OR of the current color table number and the color table number of the color of that pixel before the fill.

FORMAT:

```
-----  
OFL ;  
-----
```

USAGE:

The overlay fill command is used when a closed area is to be filled in such a way as to indicate areas of overlap or intersection between the area being filled and previously filled areas.

NOTES:

1. The area being filled must be bounded by a closed series of vectors or points, if there are any "holes" in the boundary then the fill process will "leak" through and fill undesired areas of the screen.

EXAMPLE:

DESCRIPTION	TERMINAL COMMANDS
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Draw red circle, at {250,250} radius 127.	MOV250. 250. DCL127.
Draw a filled rectangle in red.	MOV225. 325. DFR275. 350.
and in green	MOV225. 240. SEC2.
	DFR275. 265.
and in blue	MOV225. 150. SEC4.
	DFR275. 175.
and in white	MOV175. 200. SEC7.
	DFR200. 250.
Move CAP to {300,300}, inside the circle and on a black background	MOV300. 300.
Set current color to red	SEC1.
Do an overlay fill to a red boundary	OFL

OPT

programmable OPTions				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
OPT	(40	050	28

OPT

PURPOSE:

The Programmable Options (OPT) command provides a software counterpart to the "options" switches on the back panel of the terminal.

FORMAT:

```
-----  
OPT : optNo Value  
-----
```

Where:

optNo (0~7, o-type) is the option number to be changed

value (0~1, o-type) is the binary value to which the option is to be set

PROGRAMMABLE OPTIONS

Opt No.	Value	Description of the Option
2	0	Default, CR sent after coordinates in Tektronix GIN mode.
2	1	Suppresses CR in Tektronix GIN mode.
3	0	Terminal will switch to 5 x 6 font with character spacing of 7 pixels, after first G5.
3	1	No switch to 5 x 6 font.
4	0	Normal, no CR after graphic tablet "pick".

PROGRAMMABLE OPTION (Continued)

<u>Opt No.</u>	<u>Value</u>	<u>Description of the Option</u>
4	1	Send a carriage return after graphic tablet "hit" sequence (pick).
5	0	Transmit keystrokes immediately or after completion of current command, if busy.
5	1	Do not transmit keystrokes, i. e. lock out the keyboard.
6	0	Override AED command set with TEK 4014 command set where they overlap, provided that Option Switch 4 is set to 1 (up).
6	1	AED command set unconditionally override TEK 4014 command set where they overlap.
7	0	Serial data to hardcopy device.
7	1	Parallel data to hardcopy device.

USAGE:

The following are examples invoking OPT from the keyboard:

EXAMPLE:

<u>DESCRIPTION</u>	<u>TERMINAL COMMANDS</u>
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Supresses carriage return in Tektronix GIN mode	OPT2.1.
Sends a CR after graphics tablet hit (pick coordinates return sequence)	OPT4.1.

RCP

Read Current access Position

FUNCTION CODES

3 CHARS ASCII DECIMAL OCTAL HEX

RCP J 106 152 6A

RCP

PURPOSE:

The Read Current Access Position (RCP) command causes the readback of the Current Access Position (CAP) in video memory coordinates.

FORMAT:

RCP

USAGE:

RCP is used to read out the position of an image point (pixel). Upon receipt of the RCP command the AED 512/767 will transmit to the host the x,y coordinates of the CAP in the current encoding scheme (see SEN) and over the Selected Interface (see SIF) command.

The CAP is located at the center of the cursor in Graphics Interpreter Mode and at the left end of the Alphanumeric Mode (underline) cursor.

NOTES:

1. If a problem is experienced in running the manual examples, it is probably because of the following reasons:
 - a. Communications Switches on the rear of the AED 512/767 are not set properly.
 - b. Host echo of signals from the terminal has not been disabled.
 - c. For c-type encoding schemes other than decimal, the host I/O driver software is not able to pass all codes sent from the terminal.
 - d. Host does not have an input buffer that can accommodate high speed transfers consistent with the baud rates selected by COMMUNICATIONS (COM) switch settings.
2. It is recommended that modified hex encoding with the CR (carriage return) option be used.

APPLICABLE ON AED 767 ONLY

RCT*

RCT*

Read Color Table*				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
RCT	#	35	043	81

PURPOSE:

The Read Color Table (RCT) command allows the user to read the RGB values in the color table.

*Extended Function (Must be preceded by a "+" in single character mode).

FORMAT:

```
-----  
RCT sloc nloc /r1.g1.b1...rn.gn.bn/  
-----
```

Where;

sloc (0~255, o-type) is the beginning color table location from which to read.

nloc (1~256, o-type) is the number of color table locations to be read.

r1,g1,b1...rn,gn,bn (0~255, r-type) are the returned RGB values.

USAGE:

The RCT command allows the user to read back from the terminal to the host computer the current RGB values of the color table positions starting at location sloc for nloc number of locations. The number of returned bytes is equal to 3*nloc. The values returned are the same as the ones used in the SCT command to set the color table.

NOTES:

- a. If the Interpreter is in one character non-mnemonic encoding (see SEN command), then the user must prefix this command with a + (enter extended command mode). This sequence is not necessary in 3 character mnemonic encoding mode or if the host computer is capable of transmitting 8 bits of data. In the latter case the hex function code should be used.

RDA

Read Direct from AOI				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
RDA	\	47	057	2F

RDA

PURPOSE:

The Read Direct from AOI (RDA) command allows the user to sequentially read back the color table number of every pixel in the AOI (see DAI).

FORMAT:

```
-----  
RDA !  
-----
```

USAGE:

The RDA command is used to save all or portions of the graphics image on the screen for later use. This is useful for both image archival and for repeating sections of an image on the screen.

NOTES:

1. RDA and WDA (Write Direct Into AOI) are all dependent on the Area Of Interest having been defined by a DAI command and bit 5 of SUC (Set Up Counters) argument byte must be properly chosen.

RHO

Read Horizontal Origin				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
RHO	y	121	171	79

RHO

PURPOSE:

The Read Horizontal Origin (RHO) command sends, from the terminal to the host, the contents of the horizontal origin register.

FORMAT:

```
-----  
RHO ;  
-----
```

USAGE:

The RHO command is used to readback the x (horizontal) value of video memory which currently corresponds to the left edge of the screen. By doing this the user may determine how much an image has been shifted through either a pan operation from the keyboard or a Set Horizontal Origin (see SHO) command. Upon receipt of the RHO command, the AED 512/767 will transmit to the host the origin value in the current encoding scheme (see SEN) and over the Selected Interface (see SIF).

RHR

Read Horizontal Runs

FUNCTION CODES

3 CHARS	ASCII	DECIMAL	OCTAL	HEX
RHR	a	97	141	61

RHR

PURPOSE:

The Read Horizontal Runs (RHR) command is an alternative to the RDA command. It allows the user to sequentially readback the entire contents of the AOI (see DAI) in a compacted or "run length encoded" format. The information is transmitted in accordance with the Selected Interface (see SIF) and encoding (see SEN).

FORMAT:

RHR ![/n1 /color1 /n2 /color2 ... /nn /colorn] /0

Where:

n (0~255, r-type) is the number of sequential pixels
(runs) of a single color

color (0~255, r-type) is the color table number of that
color

USAGE:

For pictures which have large areas of solid colors, this command greatly decreases the amount of data needed to describe an image. Unlike Read Direct from AOI (see RDA) which transmits one value for every pixel in the AOI, RHR counts contiguous pixels of the same color and returns the number of pixels found and the color table number every time it encounters a different color pixel. It is obvious, therefore, that for images that do not contain large horizontal areas of solid color, RHR will transmit more information per image than RDA.

RHS

Read Horizontal Scan				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
RHS	t	116	164	74

RHS

PURPOSE:

The Read Horizontal Scan (RHS) command must be preceded by Define Area of Interest (DAI). RHS reads from the video memory of the terminal starting at CAP and sends to the host the color table address associated with each pixel of the entire AOI. The Terminal sends one byte per pixel in accordance with the current interface (see SIF and encoding (see SEN).

FORMAT:

RHS (/byte)

Where:

byte (0~255, p-type) is a color table location

USAGE:

The RHS command transmits a series of values defining to the host, pixel by pixel, the color which is stored in video memory and painted on the monitor screen in the Area of Interest (AOI). See the DAI command for more details of AOI.

RJP

Read Joystick Position				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
RJP	q	113	161	71

RJP

PURPOSE:

The Read Joystick Position (RJP) command will return a 9-bit approximation (0~511) of the joystick position to the host.

FORMAT:

```
-----  
RJP /x /y  
-----
```

Both x and y coordinates are sent to the host in the form of xy20.

RPX

----- Read PiXel -----				
FUNCTION CODES -----				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
RPX	Y	89	131	59

RPX

PURPOSE:

The Read Pixel (RPX) command causes the terminal to read back the color lookup table number of the pixel located at the CAP.

FORMAT:

RPX ; /value

Where:

value (0~255, r-type) is the color lookup table number of the pixel located at CAP

USAGE:

The RPX command is used to transmit to the host, the color table number of a single pixel located at CAP. If a large number of pixels in a specified area are to be read, then it is more economical in terms of transmission speeds to read the whole area through a RDA or RHR command.

EXAMPLE:

Color two adjacent pixels; one red, the next one green, against a black (default) background. Then use RPX to readback the color table addresses of the two points colored.

DESCRIPTION -----	TERMINAL COMMANDS -----
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Set Interface to Display	SIFD
Move CAP to {255,200}. Call this point P1.	MOV255.200.
Color pixel one red.	WPX1.
Move Relative one pixel to the right. Call this location P2.	MVR1.0.
Color pixel two green.	WPX2.
Read pixel (color table address) at P2. Send data to display.	RPX
Move CAP to P1	MOV 255.200.
Read pixel (color table address) at P1. Send data to display.	RPX

RRD

Read Raster Direct				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
RRD	@	64	100	40

RRD

PURPOSE:

The Read Raster Direct (RRD) command allows the user to readback color table numbers of pixels on the screen. This command does not make use of the AOI nor the current interface (see SIF) nor the current encoding scheme (see SEN). It is a DVMA operation and transmits only over the parallel interface.

FORMAT:

```
-----  
RRD i /byte  
-----
```

Where:

byte (0~255) is an 8-bit value which defines a color table location of the color of a pixel

USAGE:

The RRD command can be issued only from the host computer and transfers pixel values beginning at CAP and proceeding sequentially through n pixels over the parallel interface only. The number of pixels transmitted (n) and the memory area in the host computer into which they are read is defined by the DMA interface word count and address register.

RST

ReSeT				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
RST	0	48	060	30

RST

PURPOSE:

The Reset (RST) command initiates a full reset of the terminal, identical to the reset automatically done when the power switch is turned on or when the RESET key on the keyboard is slowly depressed twice. This command leaves all functions in default values or conditions. This command may not be issued while another command is in progress.

FORMAT:

```
-----  
RST |  
-----
```

USAGE:

The command is used when the user wishes to guarantee that all the AED 512/767 functions are in their default settings, and, that the screen is clear.

NOTES:

1. This command should not be used simply to erase the screen since it sets the color lookup table to its initial state, resets all terminal registers to their default values, and puts the terminal into Alphanumeric (TEXT) Mode.

RTP

Read Tablet Position				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
RTP	!	124	175	7D

RTP

PURPOSE:

The Read Tablet Position (RTP) command is used to read back the auxiliary digitizing tablet coordinates on demand by the host.

FORMAT:

```
-----  
RTP ! idchar status x16 y16  
-----
```

Where:

id char (0-127, o-type) is the tablet ID character previously defined by the user with the ETC command (See ETC or ETP command)

status (0-127, o-type) is the pen status byte from the digitizer tablet. When a single button cursor or stylus is used, the status byte will be:

0 for pen up (not ASCII)

1 for pen down (not ASCII)

For multiple button cursors, the status byte will be defined in the appropriate user manual for the digitizing tablet.

x16 - (0-65535, r-type) is the returned x-coordinate value.

y16 - (0-65535, r-type) is the returned y-coordinate value.

USAGE:

The RTP command is normally used with the ETP (Enable Tablet Polling) command. RTP acts as an unconditional command to the AED 512/767 to read back the digitizing cursor coordinates and pen/cursor status. If the ETC command is used, RTP will read back the status of the pen/cursor whether the pen is up or down.

RVO

Read Vertical Origin				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
RVO	z	122	172	7A

RVO

PURPOSE:

The Read Vertical Origin (RVO) command sends, from the terminal to the host, the contents of the vertical origin register.

FORMAT:

```
-----  
RVO /y  
-----
```

USAGE:

The RVO command is used to readback the y (vertical) value of video memory which currently corresponds to the top edge of the screen. By doing this the user may determine how much an image has been shifted through either a pan operation from the keyboard or a Set Vertical Origin (see SVO) command. Upon receipt of the RVO command the AED 512/767 will transmit to the origin value in the current encoding scheme (see SEN) and over the Selected InterFace (see SIF).

SAC

Set Alpha cursor Color				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
SAC	{	123	173	7B

SAC

PURPOSE:

The Set Alpha Cursor Color (SAC) command changes the alpha cursor color. (default is color 1 - red).

FORMAT:

```
-----  
SAC { color  
-----
```

Where:

color (0~255, o-type) is the color table number desired for the alpha (underscore) cursor color.

USAGE:

The SAC command is used to change the color of the alpha cursor to a different color, one that might be more visible against the current image, or indicate a new program state.

NOTES:

1. The alpha cursor is only on when the Interpreter light is off, i.e., when the terminal is in the Alphanumeric (text) Mode. Refer to XXX command.

SAP

Set Alphanumeric Parameters				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
SAP	^	94	136	5E

SAP

PURPOSE:

The Set Alphanumeric Parameters (SAP) command provides a choice between 5 x 8 and 7 x 12 pixel character fonts and for each, a choice of normal or double sized characters. It also allows the user to specify the amount the cursor is to be moved for each character and linefeed. The linking or unlinking of the CAP to the alpha cursor may also be set by this command.

FORMAT:

```
-----  
SAP i size font hspace vspace link  
-----
```

Where:

size (1/2/., o-type) if '1' indicates normal sized characters;
if '2' double size characters;
if '.' then unchanged

font (5/7/., o-type) if '5' indicates 5 x 8 character font (in 6 x 9 matrix);
if '7' indicates 7 x 9 character font (in 8 x 10 matrix);
if '.' then unchanged

hspace (0-127, o-type) number of pixels moved in the horizontal direction for each character

vspace (0-127, o-type) number of pixels moved in the vertical direction for each linefeed

link (L/U, o-type) if 'L' the Alpha cursor position is identical with CAP (default);
if 'U' the Alpha cursor position is not identical with CAP.

USAGE:

The SAP command is used to change the size of the alphanumeric characters and the amount of space left between characters (default value) There are four different character sizes:

size	font	character size	character matrix
1	5	5 x 8	6 x 9
1	7	7 x 9	8 x 13
2	5	10 x 16	12 x 18
2	7	14 x 18	16 x 26

The link parameter allows the user to unlink the CAP from the alphanumeric cursor position, this is useful when the user wishes to leave the Graphics Mode; print a message in text mode; and re-enter the Graphics Mode without changing the CAP in the graphics memory.

NOTES:

1. There are no spaces or delimiters between size, font and hspace, nor after the link parameter in o-type = D encoding (see SEN). Delimiters must be placed after h space and v space.

EXAMPLE:

DESCRIPTION	TERMINAL COMMANDS
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Set the alpha character size to be 14 x 18 pixels with 28 pixels horizontally and 36 pixel vertically between centers and link the alpha cursor to the CAP.	SAP2728.36.L
Change to the default size, 5 x 7 font pixels with 6 pixels horizontally and 9 pixels vertically between centers, and keep the alpha cursor independent of the CAP.	SAP156.9.U

SBC

Set Background Color

FUNCTION CODES

3 CHARS	ASCII	DECIMAL	OCTAL	HEX
SBC	[91	133	5B

SBC

PURPOSE:

The Set Background Color (SBC) command is used in conjunction with the ERS or Form Feed control character to erase the entire video memory to the color specified. In Alphanumeric Mode, characters backspaced over will be erased to this color. During terminal reset, the background color will be set to a color table address of zero (black in the default color table).

FORMAT:

SBC color

Where:

color is the color table address (0~255, O-type).

USAGE:

The SBC command provides the means to erase an image (alpha or graphics) and paint the entire display with a new background color. If the current background were to be retained, a reset (RST) would be used to accomplish the erasure.

EXAMPLE:

Using the default color table, put a black circle on a red background and then erase the circle.

DESCRIPTION -----	TERMINAL COMMANDS -----
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDN
Move the CAP to {100,250}.	MOV100.250.
Change background color from black (default, 0) to red (1).	SBC1.FFD
Set current color to 0 (black).	SECO.
Draw circle, black on red.	DCL50.
Set current color to 1 (red)	SEC1.
Erase circle by drawing it in background color.	DCL50.

SBL

Set BLink				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
SBL	4	52	64	34

SBL

PURPOSE:

The Set Blink (SBL) command causes one color, wherever it appears on the display, to blink to another color. The display time for each phase of the blink is specified within the command.

FORMAT:

SBL color.R.G.B.on time.off time

Where:

color is the color table of the color of pixels to blinked (0~255, O-type);

r,g,b are hue defining components of the color to which the blink will be made (0~255, O-type),

on time is the duration for which the blinked pixels will be at color; and

off time is the duration for which the blinked pixels will be at {r,g,b}; both times stated as a number of field times ie. a multiple of 1/60th second.

NOTES:

The times for both phases of the blink will change if the frame time (2 fields per frame) is other than 1/30th second, ie. 30Hz refresh rate. However, the duty cycle or ratio of the times remains unchanged.

EXAMPLE:

<u>DESCRIPTION</u>	<u>TERMINAL COMMANDS</u>
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Move CAP to {100,300}	MOV300.100.
Draw red (default) circle	DCL100.
Fill circle with yellow	SEC3. IFL
Move CAP to {100,100}	MOV100.100.
Draw green circle	SEC2. DCL100.
Fill circle with white	SEC7. IFL
Move CAP to {300,250}	MOV 300.250.
Draw blue circle	SEC4. DCL64.
Fill circle with blue	IFL
Blink yellow to red, rapidly.	SBL3. 100. 0. 0. 5. 5.
Blink white to yellow, slowly.	SBL7. 255. 255. 0. 60. 60.
Blink blue to dark grey, and at 1:10 time ratio.	SBL4. 30. 30. 30. 9. 90.

SBR

Set Baud Rate

FUNCTION CODES

3 CHARS ASCII DECIMAL OCTAL HEX

SBR b 98 142 62

SBR

PURPOSE:

The Set Baud Rate (SBR) command changes communications rates for the main (host) and aux serial interfaces from the host or from the terminal keyboard. It overrides the rates set by the back panel DIP switch settings.

FORMAT:

SBR | main aux

Where:

main and aux (0~7, o-type) are assigned codes as follows:

"0":	300 baud	"5":	9600 baud
"1":	600 baud	"6":	19200 baud
"2":	1200 baud	"7":	external/16
"3":	2400 baud	".":	Leave rate unchanged
"4":	4800 baud		

USAGE:

Soft setting of baud rate is useful in applications where a file (image) is sent to the AED 512/767 from a remote host, probably over a low speed telephone modem, then transferred from the terminal to a local host at a higher baud rate.

NOTES:

1. A full reset of the terminal from the keyboard (depressing RESET twice), or sending RST from the host causes the rate established by SBR to be reset to that of the DIP switch settings.
2. See Chapter 3 Installation, paragraph 3.5 for more detail regarding communications. The following table is repeated here for convenience.

DIP SWITCH SETTINGS

SWITCHES: 3 - 5 AUX		6 - 8 HOST	
000: 300 baud	010: 1200 baud	100: 4800 baud	110: 19200 baud
001: 600 baud	011: 2400 baud	101: 9600 baud	111: external/16

*Note: e.g. baud rate of 38,400, ext. clock should be $16 \times 38,400 = 614.4 \text{ KHZ}$

EXAMPLE:

<u>DESCRIPTION</u>	<u>TERMINAL COMMANDS</u>
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Set both host and aux baud to 300	SBR00
Set the host baud rate to 300 and the aux baud rate to 9600	SBR05

Note: Delimiters are not used.

SCC

Set Cursor Colors				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
SCC	c	99	143	63

SCC

PURPOSE:

The SCC command allows the user to blink the cursor between two colors at any desired field rate.

FORMAT:

```
-----  
SCC color1 color2 Blink-Time  
-----
```

Where:

color1 is a color number (0~255 decimal)

color2 is a color number (0~255 decimal)

Blink-time specifies the number of consecutive fields between color changes (1/30 of a sec = 1 field).

USAGE:

The SCC command is used to either reset the color of the cursor or to make the cursor blink between two colors. The rate of blinking may also be controlled.

EXAMPLE:

Blink the cursor between red and green at half second intervals.

<u>DESCRIPTION</u>	<u>TERMINAL COMMANDS</u>
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDN
Set cursor color to blink between color number 1 and color number 2 at a blink rate of 15 fields.	SCC1.2.15.

SCD

Start Command DMA				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
SCD	*	42	052	2A

SCD

PURPOSE:

The Start Command DMA (SCD) prepares the terminal to receive AED 512/767 interpreter commands over the parallel interface only.

FORMAT:

```
-----  
SCD ;  
-----
```

USAGE:

This command begins the transfer of 767/512 interpreter commands over the parallel interface. The host is freed once the transfer has begun allowing it to perform other tasks while the AED 512/767 is performing the commands. The user must place all of the commands to be executed in contiguous host memory terminated by a XCD command (must be given as a command DMA element). The user then sets the word count and buffer address registers in the parallel interface card and finally issues the SCD command.

NOTES:

1. If any commands requiring an answer back from the AED 512/767 are issued, then the interface must be set to serial (see SIF) since the parallel interface is in use.

SCP

Set Cursor Parameters

FUNCTION CODES

3 CHARS	ASCII	DECIMAL	OCTAL	HEX
SCP	J	93	135	5D

SCP

PURPOSE:

The Set Cursor Parameter (SCP) command allows the user to select the cursor shape, its spatial resolution and memory plane residence.

FORMAT:

SCP shape constraint plane(s)

Where:

shape (c-type) is X (default), + (gunsight) or L (line, which is equivalent to a full screen T square).

constraint is the spatial resolution (0~127). Default is 0.

plane(s) is the single or multiple video memory planes where the cursor resides (0~255). Default is decimal 255.

USAGE:

Set Cursor Parameters (SCP) is used to select the cursor shape, its minimal incremental movement on the screen, and in which memory plane it resides. The default condition is an X shaped cursor with single pixel minimum incremental movement which utilizes color 2 decimal, thereby, residing in memory plane 2 decimal. The cursor will appear on the screen as a green X after the terminal is powered up or fully reset, and if the cursor key is depressed or if the command (EJC or ETC) is executed.

The SCP command allows the user to select one of three shapes of cursor. The X-shape is most typically used for selecting points or pixels on the screen or for picking items from a display menu. The user may prefer a "gunsight" cursor with axes parallel to the XY coordinates. Both the X and + shaped cursors have the pixel color 0 (background) at the intersection of the cursor lines. This enables the user to accurately locate the cursor over a visible, single pixel on the screen. When the cursor is positioned over a graphic representation, every pixel from the object is saved in microprocessor memory. While the cursor is repositioned, the original pixels are retrieved from RAM and re-written to their original video memory locations. This process of saving the pixels "shadowed" by the small X or + cursor, then restoring them when the cursor is moved, occurs rapidly enough so no data is lost and the graphic representation remains intact. However, if the L attribute in the SCP command is chosen (full screen T square cursor) the total number of pixels shadowed by horizontal and vertical axes of the cursor cannot be saved and restored fast enough as the cursor is swept across the video memory. The consequence is that the original graphic representation will be eroded by the cursor movement. To avoid this object destruction, yet benefit by the T square alignment possibilities, the cursor can be assigned to a unique memory plane not otherwise occupied by the graphic representation. If the cursor color is to remain constant regardless of the background, it will be necessary to modify several color table definitions. For example, if the cursor is placed in the back memory plane (128) then the color table positions 128-255 must be defined as the desired color.

The constraint parameter in the SCP command is very useful, particularly if a grid system is displayed on the screen. Blue Line Grid (BLG) command (AED 767 only) will display a grid system of 8, 32 and 64 pixels per square. To automatically index the cursor to the nearest grid can be easily achieved by setting the constraining parameter. The constraint is enabled by choosing the number of pixels to be indexed and setting the constraint to the remaining (lower) masked bits.

AUTOMATIC INDEXING OF CURSOR POSITION

Desired resolution (index= $2^{*3}=8$ pixels)

BITS	7 6 5 4 3 2 1 0	Constraint (mask) = $2^{*0}=1.$
	Y Y Y	+ $2^{*1}=2.$
		+ $2^{*2}=4.$
		Set constraint to YYY=7.

Note: In the above, ** means exponentiation.

DESCRIPTION -----	TERMINAL COMMANDS -----
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Move to coordinates {300,300}	MOV300.300.
Draw red circle with radius 100	DCL100.
Fill circle with red	IFL
Set cursor parameter to full screen with constraining set to 8 pixel increments and the cursor set to plane 2.	SCPL7.4.
Enable joystick cursor (notice pixel ORing)	EJC
Set color table such that cursor remains white regardless of background color. (3 plane example)	SCT4.4.255.255.255. 255.255.255. 255.255.255. 255.255.255.

SCR

Send Carriage Return				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
SCR	>	62	076	3E

SCR

PURPOSE:

The Send Carriage Return (SCR) command causes the terminal to return a Carriage Return (CR, ASCII 13) to the host.

FORMAT:

SCR

USAGE:

The SCR command is used by the host to validate that the serial input buffer of the terminal is empty and the terminal has completed the processing of serially transmitted data, before changing to parallel transmission. The terminal responds to the SCR by returning a CR to the host. This is just one typical use of the SCR command.

SCS

Set Console Status				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
SCS	'	96	140	60

SCS

PURPOSE:

The Set Console Status (SCS) command allows the host computer to accomplish the same functions as those of the special function keys on the top row of the keyboard. Most familiar to the user would be "joystick rate" and "joystick vernier". All are listed below.

FORMAT:

SCS byte

Where:

byte is an encoded numeral setting the desired bits from the following table:

BIT		BIT	
---		---	
0:	joystick vernier	4:	reverse video enable
1:	joystick rate	5:	scroll disable
2:	hard copy enable	6:	lower case enable
3:	trace enable		

USAGE:

Sending the bits of byte, following the SCS function code, has the same effect as depressing the special function key. In SEN3DDDN encoding, for example,

- a. SCS1. Enables the joystick vernier (bit 0 set to 1)
- b. SCS32. Disables the scroll (bit 5 set)
- c. SCS5. Enables hard copy and joystick vernier (bit 0 and 2 set)
- d. SCS127. Equivalent to depressing all seven keys with special function labels (bits 0 through 6 all set)
- e. SCS0. Resets any and all bits previously set, restores the functions listed under format to their default conditions.

In SEN1888N encoding the same examples would be:

- a. Accent grave CTRLA Enables joystick vernier
- b. Accent grave Space bar Disables the scroll
- c. Accent grave CTRL E Enables hard copy and joystick vernier
- d. Accent grave RUBout Equivalent to depressing all keys
- e. Accent grave CTRL@ Resets all keys

*Note: CTRLA is simultaneous depression of the CTRL and A keys. This produces an ASCII encoded, decimal 1 (see Chapter 4).

SCT

Set Color Table				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
SCT	K	75	113	4B

SCT

PURPOSE:

The Set Color Table (SCT) command is used to expand or completely redefine the default color table.

FORMAT:

```
-----  
K addr n (r1 g1 b1)  
-----
```

Where:

addr is the first of a series of consecutive addresses to be redefined, (0~255, o-type);

n is the number of consecutive addresses whose contents (color intensities or hues) are to be changed, (0~255, o-type); and

(r1,g1,b1) is the triad(s) of color intensity values to be supplied for each address, each element (0~255 o-type).

The color lookup table will actually be updated during the next vertical blanking interval after the complete instruction is executed.

USAGE:

Set Color Table (SCT) may be used to expand the default color table, perhaps leaving unchanged the color (hue) definitions of addresses (color numbers) 0 through 7, but changing the default black assigned to color numbers 8 through 255. Or, the entire "address versus hue" mapping may be redefined.

Color Number or Address	Color (Hue)
0	Black
1	Red
2	Green
3	Yellow
4	Blue
5	Magenta
6	Cyan
7	White
8 - 255*	Black

*Note: Color in these locations programmed by user will default to black on initialization.

EXAMPLE:

Change color location numbers 11 through 13 as follows, leaving unchanged the default definitions of number 0 through 10.

```
RESET RESET LOCAL
ESC CAPS LOCK
SET ENCODING
SEN3DDDN
SCT11. 3. 200. 100. 0. 200. 0. 0. 150. 150. 150.
```

The resulting partial color table is:

Color No.	R	G	B	
11	200	100	0	Dark Red
12	200	0	0	Bright Red
13	150	150	150	Grey

SDA

----- Stop Direct Access -----				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
SDA	ESC	27	033	1B

SDA

PURPOSE:

The Stop Direct Access (SDA) command halts the reading or writing of memory using DVMA data transfers. See RRD for reading and WRD for writing.

FORMAT:

SDA

USAGE:

When using the SDA command you may stop the reading or writing of memory by utilizing the DMVA data transfers.

SEC

SEt current Color				

FUNCTION CODES				

3 CHARS	ASCII	DECIMAL	OCTAL	HEX

SEC	C	67	103	43

SEC

PURPOSE:

The Set Current Color (SEC) command determines the color for drawing vectors, characters (alphanumeric or special font), circles, and filled areas. The default value for the current color is 1 (red).

FORMAT:

SEC | color

Where:

color (0~255, o-type) is the color table number of the current color

USAGE:

SEC sets the drawing color when other than the default color is desired. During power-up, or upon the reset (RST) command, the current color is set to 1, which is red in the default color table.

EXAMPLE:

<u>DESCRIPTION</u>	<u>TERMINAL COMMANDS</u>
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Set current color to 3 of the color table now in effect. If that is the default table, then the next figure drawn will be yellow (3)	SEC3.
Move CAP to {100,250}	MOV100.250
Draw yellow circle centered at {100,250} with radius 50 pixels.	DCL50.
Set current color 4 (blue).	SEC4.
Draw blue circle inside the yellow one, radius 30.	DCL30.
Set current color 1 (red)	SEC1.
Draw red circle inside the yellow circle, radius 10.	DCL10.

SEN

Set ENcoding scheme				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
SEN	G	71	107	47

SEN

PURPOSE:

The Set Encoding Scheme (SEN) command provides the capability to change the way function codes and operands of subsequent commands will be entered into the terminal and returned to the host.

FORMAT:

```
-----  
SEN : ftype  otype  rtype  ctype  ptype  
-----
```

Where:

ftype (1/3/., o-type) is the encoding scheme for function codes (commands),

otype (7/8/H/M/D/., o-type) is the encoding scheme for numeric operands

rtype (7/8/H/M/D/., o-type) is the encoding scheme for returned (from terminal) operands

ctype (7/8/H/M/D/., o-type) is the encoding scheme for coordinat (to or from terminal) operands

pctype (7/8/N/M/., o-type) is the encoding scheme for pixel operands

Each of the above five arguments of the SEN function have their possible values described in the following.

Function code ftype

- ftype = "1" Abbreviated function codes. One character, ASCII, non-mnemonic. (Default)
- ftype = "3" Symbolic function codes. Three character, 3 CHARS, mnemonic.
- ftype = "." Leave ftype unchanged.

Numeric operand otype

- otype = "7" 7-bit binary
- otype = "8" 8-bit binary. (Default)
- otype = "H" 2-digit ASCII representation of hexadecimal digits
- otype = "M" 2-digit modified ASCII representation of hexadecimal. Hex digit (0123456789ABCDEF) corresponds to (0123456789:;<=>?).
- otype = "D" Variable length decimal. Anything but decimal digits and numeric sign acts as a terminator.
- otype = "." Leave otype unchanged

Returned operand rtype

- rtype = "7" 7-bit binary
- rtype = "8" 8-bit binary (Default)
- rtype = "H" Hexadecimal
- rtype = "M" Modified hexadecimal
- rtype = "D" Variable length decimal, terminated by CR
- rtype = "." Leave rtype unchanged

Coordinates operand ctype

ctype = "7" 7-bit binary. See Note 1
ctype = "8" 8-bit binary. (Default) See Note 1
ctype = "H" 5 character hexadecimal. See Note 1
ctype = "M" 5 character modified hexadecimal. See Note 1
ctype = "D" Variable length decimal, x followed by y.
ctype = "." Leave ctype unchanged.

Pixel operand p type

ptype = "7" Seven pixels per byte, one bit per pixel.
ptype = "8" Eight pixels per byte, one bit per pixel.
ptype = "N" Normal. One pixel per byte. (Default)
ptype = "M" Modified. Seven least significant bits refer to seven most significant memory planes. One pixel per byte.
ptype = "." Leave ptype unchanged.

NOTES:

ENCODING OF COORDINATES

1. Encoding of coordinate pairs (e.g. the xy20 argument of MOV function)

a. 8-bit Binary (ctype = "8")

1st Byte:	X11	X10	X9	X8	Y11	Y10	Y9	Y8
2nd Byte:	X7	X6	X5	X4	X3	X2	X1	X0
3rd Byte:	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0

b. 7-bit Binary (ctype = "7")

1st byte: xx X9 X8 X7 Y10 Y9 Y8 Y7
2nd byte: xx X6 X5 X4 X3 X2 X1 X0
3rd byte: xx Y6 Y5 Y4 Y3 Y2 Y1 Y0

c. Hexadecimal (ctype = "H")

1st Character: X9 X8 Y9 Y8
2nd Character: X7 X6 X5 X4
3rd Character: X3 X2 X1 X0
4th Character: Y7 Y6 Y5 Y4
5th Character: Y3 Y2 Y1 Y0

2. Encoding of 16 bit integers (e.g. address in JUS)

Hexadecimal rtype = "H")

ENCODING DOUBLE PRECISION UNSIGNED NUMBERS

8-bit Binary (otype or rtype = "8")

1st byte: N15 N14 N13 N12 N11 N10 N9 N8
2nd byte: N7 N6 N5 N4 N3 N2 N1 N0

7-bit binary mode (otype or rtype "7")

1st byte: xx N13 N12 N11 N10 N9 N8 N7
2nd byte: xx N6 N5 N4 N3 N2 N1 N0

Hexadecimal or Modified Hexadecimal (4 characters)

1st Char: N15 N14 N13 N12 3rd Char: N7 N6 N5 N4
2nd Char: N11 N10 N9 N8 4th Char: N3 N2 N1 N0

EXAMPLE OF THE 7-BIT ARGUMENT ENCODING PROCESS

Let's encode the x, y coordinates 100,200. This is argument encoding.

PROCEDURE: (EXAMPLE BELOW)

Step:

1. Display the "base-10" coordinates to be encoded.
2. Convert the coordinates to the hexadecimal (base-16) system.
3. For tutorial purposes, convert hex to binary and relate bits to the terminology of the SEN command in (Refer to "c-type argument "and" Note 1b under ENCODING OF COORDINATES" at the SEN command). Assemble bit combinations to form bytes.
4. Convert bytes B1, B2, and B3 to ASCII characters for keyboard entry, or to the programmers choice (hex, octal, etc) for host code generation. THIS IS THE DESIRED ENCODING.

PROCEDURE STEP	X	Y
1.	100/10	200/10
2.	$0640/16 = (0)16^2 + (6)16^1 + (4)16^0$ $100/10 = 96/10 + 4/10$	$0C8/16 = (0)16^2 + (12)16^1 + (8)16^0$ C is hex for 12 $200/10 = 192/10 + 8/10$
3.	$0/16 = 0000_2 = X11X10X9X8$ $6/16 = 0110_2 = X7X6X5X4$ $4/16 = 0100_2 = X3X2X1X0$	$0/16 = 0000_2 = Y11Y10Y9Y8$ $C/16 = 1100_2 = Y7Y6Y5Y4$ $8/16 = 1000_2 = Y3Y2Y1Y0$

From the SEN command description:

1st Byte, B1 - xx X9 X8 X7 Y10 Y9 Y8 Y7 = 00000001 = 1/10

X-Nibble Y-Nibble

where xx (don't care) bits are made zero.

2nd Byte, B2 = xx X6 X5 X4 X3 X2 X1 X0 = 01100100 = 100/10

Byte

3rd Byte, B3 = xx X6 X5 X4 X3 Y2 Y1 Y0 = 01001000 = 72/10

4. B1 = "CTRL A" = 01/16 = 001/8 = 1/10
B2 = "d" = 64/16 = 144/8 = 100/10
B3 = "H" = 48/16 = 110/8 = 72/10

So the result is that 100,200 encodes to the sequence "CTRL A"
"d" "H" in ASCII equivalent characters or keystrokes.

EXAMPLE OF THE 8-BIT ARGUMENT ENCODING PROCESS

Suppose it's required that we encode the coordinates 100,250.

PROCEDURE:

Step:

1. Display the "base-10" coordinates to be encoded
2. Convert the coordinates to the hexadecimal (base-16) system.
3. For tutorial purposes, convert hex to binary and relate bits to the terminology of the SEN command in. (Refer to "c-type" Argument under FORMAT and Item 1a under ENCODING OF COORDINATES at the SEN command). Assemble bit combinations to form bytes.
4. Convert bytes B1, B2, and B3 to ASCII characters for keyboard entry, or to the programmers choice

PROCEDURE STEP	X	Y
1.	100/10	250/10
2.	$100/16 = (0) \cdot 16^2 + (6) \cdot 16^1 + (4) \cdot 16^0$ $100/10 = 96/10 + 4/10$	$250/16 = (0) \cdot 16^2 + (15) \cdot 16^1 + (10) \cdot 16^0$ F is hex for 15 A is hex for 10 $250/10 = 240/10 + 10/10$
3.	Bit Labels $0/16 = 0000 = X11X10X9X8$ $6/16 = 0110 = X7X6X5X4$ $4/16 = 0100 = X3X2X1X0$	$0/16 = 0000 = Y11Y10Y9Y8$ $F/16 = 1111 = Y7Y6Y5Y4$ $A/16 = 1010 = Y3Y2Y1Y0$

From the SEN command description:

1st Byte, B1 = X11X10X9X8 Y11Y10Y9Y8 = 00000000 = 0/10
Upper Nibble Upper Nibble
of X of Y

2nd Byte, B2 = X7X6X5X4X3X2X1X0 = 01100100 = 100/10

3rd Byte, B3 = Y7 Y6 Y5 Y4 Y3 Y2 Y1 Y0 = 11111010 = 250/10

From the ASCII chart in Chapter 4,

B1 = "Nul" = 00/16 = 000/8 = 0/10
B2 = "d" = 64/16 = 144/8 = 100/10
B3 = see = FA/16 = 372/8 = 250/10
note
below

Note: Because B3 > 127/10 it, is not available in a single ASCII character. So there's no key on the terminal keyboard with which this byte may be entered. Therefore, the command MOV 100.250. must be encoded differently if you want to enter it through the keyboard. Of course, this encoding is the desirable one for host transmission of 8 or 16 bit words to the terminal.

The result is that 100.250 encodes to the following bit stream:

0 0 0 0 0 0 0 0	0 1 1 0 0 1 0 0	1 1 1 1 1 0 1 0
1st Byte	2nd Byte	3rd Byte

EXAMPLE OF ARGUMENT ENCODING IN HEX

This time, we'll use x, y coordinates 511, 483 - the upper right corner of a default condition AED 512 display - as an example.

PROCEDURE (Example below)

Step:

1. Display the "base-10" coordinates to be encoded.
2. Convert the coordinates to the hexadecimal (base-16) system.
3. For tutorial purposes, convert hex to binary and relate bits to the terminology of the SEN command. (Refer to "c-type) argument and Note 1c under ENCODING OF COORDINATES at the SEN command). Assemble bit combinations to form bytes.

PROCEDURE STEP	X	Y
1.	511/10	483/10
2.	$511/16 = (1)16^2 + (15)16^1 + (15)16^0$ F is hex for 15 $511/10 = 256/10 + 240/10 + 15/10$	$483/16 = (1)16^2 + (14)16^1 + (4)16^0$ E is hex for 14 $483/10 = 256/10 + 244/10 + 4/10$
3.	Bit Labels $1/16 = 0001 = 0 0 X9 X8$ $F/16 = 1111 = X7 X6 X5 X4$ $F/16 = 1111 = X3 X2 X1 X0$	$1 = 0001 = 0 0 Y9 Y8$ $E = 1110 = Y7 Y6 Y5 Y4$ $4 = 0100 = Y3 Y2 Y1 Y0$

From the SEN command description:

ASCII

1st Character, C1 = X9 X8 Y9 Y8 = 0101 = 5/10 = 5/16 = 53
2nd Character, C2 = X7 X6 X5 X4 = 1111 = 15/10 = F/16 = 70
3rd Character, C3 = X3 X2 X1 X0 = 1111 = 15/10 = F/16 = 70
4th Character, C4 = Y7 Y6 Y5 Y4 = 1110 = 14/10 = E/16 = 69
5th Character, C5 = Y3 Y2 Y1 Y0 = 0100 = 4/10 = 4/16 = 52

4. Having previously put the AED terminal into Hex Mode (for example, with a G3HHN keystroke sequence), let's move the CAP to (511,483) and write a pixel. The keystrokes required at the AED terminal or at the host's local terminal are:

MOV5FFE4WPX

The result is that 511,483 encodes, from hex, to the following bit stream: *

CHARACTERS

"F" = 01000110 "E" = 01000101

"5" = 00110101 "F" = 01000110 "4" = 00110100

1st 2nd 3rd 4th 5th

*Data packed 1 character/byte

EXAMPLE OF ARGUMENT ENCODING IN MODIFIED-HEX

This time, we'll use x,y coordinates 511,483 - the upper right corner of a default condition AED 512 display - as an example.

PROCEDURE: (Example Below)

Step:

1. Display the "base-10" coordinates to encoded.
2. Convert the coordinates to the hexadecimal (base-16) system.
3. For tutorial purposes, convert hex to binary and relate bits to the terminology of the SEN command. (Refer to "c-type" argument and Note 1c under ENCODING OF COORDINATES at the SEN command). Assemble bit combinations to form bytes.
4. Send the appropriate ASCII encoded characters to the terminals microprocessor by a) having the terminal in Modified-Hex Mode and sending a keystroke sequence of modified-hex characters, or b) sending from the host the ASCII equivalents of the hex characters.

PROCEDURE STEP	X	Y
1.	511/10	483/10
2.	$15 \times 16^2 + 15 \times 16^1 + 15 \times 16^0$ F is hex for 15 $511/10 = 256/10 + 240/10 + 15/10$	$14 \times 16^2 + 14 \times 16^1 + 4 \times 16^0$ E is hex for 14 $483/10 = 256/10 + 244/10 + 4/10$
3.	Bit Labels $1/16 = 0001 = 0 \ 0 \ X9 \ X8$ $F/16 = 1111 = X7 \ X6 \ X5 \ X4$ $F/16 = 1111 = X3 \ X2 \ X1 \ X0$	$1 = 0001 = 0 \ 0 \ Y9 \ Y8$ $E = 1110 = Y7 \ Y6 \ Y5 \ Y4$ $4 = 0100 = Y3 \ Y2 \ Y1 \ Y0$

From the SEN command description:

						MODIFIED	ASCII
						HEX	
1st Character, C1	=	X9 X8 Y9 Y8	=	0101	=	5/10	= 5 = 53/10
2nd Character, C2	=	X7 X6 X5 X4	=	1111	=	15/10	= = 63/10
3rd Character, C3	=	X3 X2 X1 X0	=	1111	=	15/10	= = 63/10
4th Character, C4	=	Y7 Y6 Y5 Y4	=	1110	=	14/10	= > = 62/10
5th Character, C5	=	Y3 Y2 Y1 Y0	=	0100	=	4/10	= 4 = 52/10

4. Having previously put the AED terminal into Modified-Hex Mode (for example, with a G3MMN keystroke sequence), let's move the CAP to (511,483) and write a red pixel. The keystrokes required at the AED terminal or at the host's local terminal are:

MOV5??4WPX1.

The result is that 511,483 encodes, from modified-hex, to the following bit stream: *

CHARACTERS

" " = 00111111

">" = 001111110

"5" = 00110101

"B" = 01000110

"4" = 00110100

1st

2nd

3rd

4th

5th

*Data packed 1 character/byte

SHO

Set Horizontal Origin				
COMMAND CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
SHO	f	102	146	66

SHO

PURPOSE:

The Set Horizontal Origin (SHO) command allows the user to set the address in video memory where the horizontal raster scan will begin.

FORMAT:

```
-----  
SHO : value  
-----
```

Where:

value (0~511/1023, o-type) is the x (horizontal) coordinate in video memory which is to correspond to the left edge of the screen.

USAGE:

The SHO command is used to change the horizontal reference of the AED 512/767 display. Together with the SVO command, the user may change the orientation of the AED 512/767 display. Also see BSO.

NOTES:

1. The default coordinate is 0. Change in the default is executed during the next vertical retrace interval.

EXAMPLE:

Draw 100 pixel radius circle and with the SH0 command move screen display to the left.

DESCRIPTION -----	TERMINAL COMMANDS -----
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Move the CAP to {400,100}	MOV400.200.
Draw a circle with a radius of 100 pixels	DCL100.
Move display to left	SH0200.

SIF

Select InterFace				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
SIF	H	72	110	48

SIF

PURPOSE:

The Select Interface (SIF) command selects either the S (Serial) interface, the P (Parallel) interface, the D (Display) interface or M (Mixed) interface for communications with the host in subsequent commands.

FORMAT:

```
-----  
SIF i char  
-----
```

Where:

char (S/P/M/D, o-type) is the interface type to be used

USAGE:

There are 4 modes by which the user may communicate with the AED 512/767 from the host. The first is Serial. In this mode, commands are issued and readbacks received over the host RS-232C interface. This mode is used when communicating over a short serial cable to a local host or to a remote host over a telephone line with a modem. The second is Parallel. In this mode all communications are performed over the 50 pin parallel connector and requires a special hardware interface be installed in the host computer. Generally the host computer is not more than 50 feet away from the AED 512/767 in this mode. The third mode is Mixed. In this mode, keystrokes are transmitted over the serial interface while returned arguments are transmitted over the parallel interface. The last mode is Display which causes returned data to be sent to the 512/767 display at the CAP, i. e., if RTP was issued by the host, the terminal would display the actual XY coordinates of the tablet stylus position.

NOTES:

1. The SIF command affects only returned arguments. Commands may be issued over the serial or parallel interfaces regardless of the current setting of the SIF command.

SKS

----- Send KeyStroke -----				
FUNCTION CODES -----				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
SKS	6	54	066	36

SKS

PURPOSE:

The Send Keystroke (SKS) command is sent by the host when it is ready to process a keystroke character from the terminal through the parallel port.

FORMAT:

SKS : /code

Where:

code (0~127, o-type) is the encoded keystroke (key character)

USAGE:

The SKS command only works when the interface has been previously set to P (Parallel) with the SIF command. When SKS is sent from the host, it causes the most recent keystroke to be returned to the host. If several keys have been struck since the last SKS, the last one will be returned. If no key has been struck since the last SKS command, then the terminal will hang (wait) with its busy indicator light on, until a key is struck and its code is immediately returned over the parallel interface.

SLS

Set Line Style				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
SLS	1	49	61	31

SLS

PURPOSE:

The Set Line Style (SLS) command allows the user to generate a variety of single width (weight) line patterns.

FORMAT:

```
-----  
SLS | pattern  scale  
-----
```

Where:

pattern (0~255, o-type) is an 8-bit binary number, specifies a basic broken line pattern with;

- 1 = MARK [visible]
- 0 = SPACE [invisible]

scale (0~255, o-type) allows the basic 8 bit pattern to be scaled up in length by a factor as follows:

Argument	Length Factor
255	1 (Single Weight)
85	2 (Double Weight)
17	4 (Quadruple Weight)
1	8 (Eight Times Weight)

USAGE:

The SLS command provides a basis for visual differentiation between lines (vectors) of the same color. It is used in complex graph drawing and in drafting/CAD/CAM applications, where certain broken line slashing patterns are common. For each one (1) or zero (0) bit in the pattern a pixel times the length factor in scale is turned on or off. The pattern is repeated until the vector is completely drawn. The SLS command should not be used for drawing polygons which are to be filled because spaces in the polygon outline will cause the fill to leak through boundaries.

EXAMPLE:

DESCRIPTION -----	TERMINAL COMMANDS -----
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Set current color to green	SEC2.
Move CAP to {240,400}	MOV240.400.
Draw vector 32 pixels long, (4 patterns) to {271,400}	DVA271.400.
Move CAP to {240,395}	MOV240.395.
Set line style, 01100110* (Single Weight)	SLS102.255.
Draw vector to {271,395}	DVA271.395.
Move CAP to {240,390}	MOV240.390.
Set line style, 10101010* (Single Weight)	SLS170.255.
Draw vector to {271,390}	DVA271.390.
Move CAP to {240,385}	MOV240.385.
Set line style, 00001111* (single Weight)	SLS15.255.
Draw vector to {271,385}	DVA271.385.
Move CAP to {240,380}	MOV240.380.
Set line style as above but double the weight (length of the broken pattern)	SLS15.85.
Draw vector to {271,380}	DVA271.380.

* derivation of pattern values:

(0 = pixel(s) off, 1 = pixel(s) on)

base:	2	10	16
	-----	---	--
01100110	==	102	== 66
10101010	==	170	== AA
00001111	==	15	== 0F

APPLICABLE ON AED 767 ONLY

Select Pattern Fill

FUNCTION CODES

3 CHARS ASCII DECIMAL OCTAL HEX

SPF " 34 042 22

SPF

SPF

PURPOSE:

The Set Pattern Fill (SPF) command allows the user to set the pattern to be used in subsequent filled rectangles. See DSP command.

FORMAT:

SPF npat

Where

npat (0~15, o-type) is the pattern number to be used

USAGE:

The SPF command is used to specify the stipple fill pattern to be used in all subsequent DFR (Draw Filled Rectangle) and FRR (draw Filled Rectangle Relative) commands. The pattern must have previously been defined by the DFP command. If a value of 0 (zero) is used for npat then stipple fill is disabled and all subsequent filled rectangles will use solid fill.

SRM

Set Read Mask				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
SRM	M	77	115	4D

SRM

PURPOSE:

The Set Read Mask (SRM) command allows video data to be read from a selected subset of the the memory planes contained in the AED 512/767 hardware configuration. Therefore a selected subset of color lookup table locations is available.

FORMAT:

```
-----  
SRM i mask0 mask1 mask2 mask3  
-----
```

Where:

mask (0~255, o-type) is an 8 bit byte which is logically ANDed with video memory data bytes to determine color lookup table locations for pixels being displayed.

512 ONLY:	mask0	determines the color lookup table (CLT) locations which are addressable when the video sweep origin is at [0,483 (or 512 for CCIR proms)]; this is the default condition display.
-----	mask1	determines which CLT locations are addressable to provide the horizontal wrap-around image.
	mask2	determines which CLT locations are addressable to provide the vertical wrap-around image.
	mask3	determines which CLT locations are addressable to provide the diagonal wrap-around obtained when both horizontal and vertical origin registers are changed from their default values of 0 and 483 (or 512), respectively.
767 ONLY:	mask0-3	determine which CLT locations are addressable in the lower left, lower right, upper left and upper right 512x512 quadrants of video memory. If mask1=mask2=mask3=0 then mask1=mask2=mask3=mask0

USAGE:

The SRM command restricts the color table locations which may be accessed when the image in video memory is displayed on the screen by "filtering" the color table locations stored in video memory. This is done by ANDing the color table location in video memory for a pixel with the read mask and using the resultant value as the color table location for the corresponding pixel on the screen. The value in video memory is unchanged by this operation.

EXAMPLE:

Draw the sketched pattern, with display (sweep) origin at (0,483) of video memory [default condition] and in the colors shown. Observe the effects of using read masks, with and without panning.

TERMINAL DESCRIPTION -----	COMMANDS -----
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Draw border 10 pixels in red	MOV0.0. DFR511.483. SECD.MOV 10.10. DFR502.473.
Set color to green, draw diagonals	SEC2. MOV0.0. DVA511.483. MOV0.483. DVA511.0.
Set color to blue, draw crosshairs	SEC4. MOV0.242. DVA511.242. MOV255.482. DVA255.0.
Set Read Mask to transparency (default condition)	SRM255.255.255.255.

(EXAMPLE CONTINUED ON NEXT PAGE)

Move (pan) image; use these commands or manual (joystick) control.

SH0400.
SV0100.
BS00.483.

Note: That the Full color (RGB) image is visible in all wrap-arounds; H, V, and D. Define the above three commands or user action as PAN SEQUENCE

Set Read Mask to the second entry in tabulation under Usage, repeat above PAN SEQUENCE commands.

SRM255.1.1.1.

Continue through the examples tabulated under Usage and repeat PAN SEQUENCE, noting the image effects described there.

SRM255.2.0.0.
PAN SEQUENCE

SRM255.0.2.0.
PAN SEQUENCE

SRM255.0.0.2.
PAN SEQUENCE

SRM255.0.0.0.
PAN SEQUENCE

SRM5.2.0.0.
PAN SEQUENCE

Return read mask to transparency.

SRM255.255.255.255.

SSE

----- Set Stack End -----				
FUNCTION CODES				

3 CHARS	ASCII	DECIMAL	OCTAL	HEX

SSE	}	125	175	7D

SSE

PURPOSE:

The Set Stack End (SSE) command is used to set the upper end (highest address) of RAM space to be allotted to a "scratch pad" memory stack. Memory demand reaching the location address will cause the terminal's microprocessor to halt execution of that command.

FORMAT:

```
-----  
SSE | addr\16  
-----
```

Where:

addr (0~65535, o-type) is the 16-bit address defining the upper limit of the stack memory space

USAGE:

1. The SSE command will be used along with the LMR command to allocate RAM space between the top of addressable RAM and the RAM space used by the AED 512/767 interpreter.

The JUS (Jump to User Subroutine) will be executed to transfer from a host program to a subroutine stored and executed in the terminal.

2. Depending on the amount of RAM storage space required for the subroutine, n bytes, the remaining RAM space is used as Last In First Out (LIFO) stack. The stored subroutine is protected from over-writing by proper choice of the address argument of the SSE (Set Stack End) command. If the stack (including storage requirements of DSF {Define Special Font} and {Define Special Key}) gets too large, the SSE address is reached. The microprocessor then will exit whatever routine it is executing, return the pointer to the start of that routine, and halt further execution of instructions.

STD

Set Turnaround Delay				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
STD	'	39	047	27

STD

PURPOSE:

The Set Turnaround Delay (STD) command holds off transmission of serial data until delay/60 seconds after receipt from the host of the last serial character of the requesting command.

FORMAT:

```
-----  
STD ; delay  
-----
```

Where:

delay (0~65535, o-type) is the time delay in returning data from the terminal to the host, in 60ths of a second.

USAGE:

This command is used in applications such as timesharing, where there is a time lapse between the host issuing a command and the host being ready to receive the answer.

NOTES:

1. This command applies only to values returned over the serial interface (see SIF).

STP*

Set Tablet Parameters					
3	CHARS	ASCII	DECIMAL	OCTAL	HEX
STP	'		40	50	86

STP*

PURPOSE:

The Set Tablet Parameters (STP) command is used in conjunction with the ETC command. This function controls the selection of multiple or single stylus hits.

*Extended function (must be preceded by a "+" in single character mode).

FORMAT:

```
-----  
STP | code  
-----
```

Where;

code = 0 single stylus hits (default)

code = 1 multiple stylus hits

USAGE:

The bit pad is usually used with single stylus hits. However, multiple stylus hits can be used if bit pad data is to be sent on pen up and pen down hits.

APPLICABLE FOR AED 767 ONLY

STW*

Set Tektronix Window					

3	CHARS	ASCII	DECIMAL	OCTAL	HEX

STW	41)	51	87	

STW*

PURPOSE:

The Set Tektronix Window (STW) command establishes either 9 bit or 10 bit resolution for Tektronix emulation by the AED 767. The STW command is issued before the SUB or GS commands are issued and following a full RESET of the terminal.

*Extended function (must be preceded by a "+" in single character mode).

FORMAT:

STW ! code

Where:

code (0/1, o-type) code = 0 (default) establishes 9-bit resolution (0~511) so the screen displays the full memory contents. Code 1 establishes 10-bit resolution (0~1023) so the screen must be panned across the full memory content.

SUB

Emulate Tektronix GIN Mode				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
See				
Note 1	CRTL Z	26	032	1A

SUB

PURPOSE:

The Emulate Tektronix Graphics Input mode command turns on the graphics cursor and enables its positioning by the joystick. When the user presses a key on the AED 512/767 the CAP is returned to the host in Tektronix 4010 format. This command is only valid in ftype = 1 (single character command) mode.

FORMAT:

(simultaneous depression of CTRL and Z keys)

USAGE:

1. Position cursor to the desired locations
2. Hit any key
3. The code for this key is transmitted, followed by a terminal status byte, then followed by the cursor coordinates encoded as follows:

Note: 1. This command is valid only in ftype=1, single character encoding. Also see SEN command.

ENCODING OF COORDINATES

Tektronix compatible coordinates from host are sent in a sequence of from one to four characters following the first (CTRLZ) character.

512 NORMAL

	Bit	7	6	5	4	3	2	1	0
2nd Character	hi. x	XX	0	1	X8	X7	X6	X5	X4
3rd Character	lo. x	XX	1	1	X3	X2	X1	X0	0
4th Character	hi. y	XX	0	1	Y8	Y7	Y6	Y5	Y4
5th Character	lo. y	XX	1	0	Y3	Y2	Y1	Y0	0

The coordinates are followed by CR (13) unless programmable option of opt No. 2 has value = 1. See OPT command.

767 NORMAL OR 512 SUPERDAM (Firmware Version 88 and up)

	Bit	7	6	5	4	3	2	1	0
2nd Character	hi. x	XX	0	1	X9	X8	X7	X6	X5
3rd Character	lo. x	XX	1	1	X4	X3	X2	X1	X0
4th Character	hi. y	XX	0	1	Y9	Y8	Y7	Y6	Y5
5th Character	lo. y	XX	1	0	Y4	Y3	Y2	Y1	Y0

NOTES:

1. The terminal command protocol has been structured such that the AED 512/767 is fully compatible with the non-raster Tektronix 4010-4014 series terminals. It is possible to run unmodified Plot-10 software to produce an appropriate display on the AED 512/767 consisting of vector graphics with alphanumeric legends. Graphics input (GIN) mode will also be correctly emulated to allow the operator to use the AED 512/767's joystick and cursor without software modification. Additional commands to control unique terminal functions (e.g. color definition), can be integrated into most implementations without modification to existing host graphics software.
2. The AED 512/767 will perform complete 4010-4014 emulation (with or without the Enhanced Graphic Option, Option 34), including multiple line styles, incremental plot, random point plotting, and multiple character sizes.

3. For maximum flexibility, Tektronix emulation modes can be entered from either alphanumeric mode or from the graphics interpreter.
4. The STW command will change the magnitude of the coordinates. See STW.

SUC

Set Up Counters for DVMA				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
SUC	?	63	077	3F

SUC

PURPOSE:

The Set Up Counters (SUC) for Direct Video Memory Access (DVMA) commands defines the spatial arrangement of incoming data for placement into video memory.

FORMAT:

SUC : byte

Where:

byte (0~255, o-type) is defined as follows:

BIT

- 0: X auto-increment (default = 1) [Least significant bit]
- 1: Y auto-increment (default = 0)
- 3: X up counter (left to right) (default = 1)
- 4: Unlinked counters. Overflow from one counter will not increment the other. (default = 0)
- 5: 16-bit transfers, (as opposed to 8-bit. (default = 1)
- 6: 0 (Reserved)
- 7: 0 (Reserved) [Most significant bit]

Only bit 5 of byte is used (possibly changed from default) for Area of Interest (AOI) transfers.

APPLICABLE ON AED 512 ONLY

SUP

SUP

SUPERoam				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
SUP	-	45	55	2D

PURPOSE:

The Superoam (SUP) command automatically enables or disables the reconfiguration of video memory to a 1024 x 1024 pixel area viewable through a window of up to 512 x 483 or 512 pixels. The increase in write/read area is accomplished at the cost of a reduced number of simultaneously displayable colors. THE AED 767 DOES NOT SUPPORT SUPEROAM.

FORMAT:

SUP | arg

Where:

arg (0/1, o-type) if 1 enables 1024 x 1024 SUPEROAM mode,
and if 0 disables it

USAGE:

SUPEROAM refers to the AED 512 capability of restacking its memory planes (4 planes into 1024 x 1024 x 1, and 8 planes into 1024 x 1024 x 2 planes deep) and to adjust its display origin registers to allow a continuous roam through the enlarged video memory address space.

After enabling SUPEROAM, the terminal will interpret coordinates in the range 0 ~ 1023, rather than 0 ~ 511. All commands referring to memory planes will have their operands clipped to the least significant two bits.

NOTES:

1. Refer to the SWM and SRM commands for more information on how to read/write from various memory planes.

EXAMPLE:

This example shows how 1024 x 1024 coordinates are implemented on the AED 512.

<u>DESCRIPTION</u>	<u>TERMINAL COMMANDS</u>
INITIALIZE TERMINAL	RESET RESET LOCAL ESC
SET ENCODING	SEN3DDDN
ENABLE SUPERDAM	SUP1.
Move to {0,0}, draw to {1023,1023}	MOV0. 0. DVA1023. 1023.
Change color from red to green	SEC2.
Move to {0,1023}, draw to {1023,0}	MOV0. 1023. DVA1023. 0.
Change color to yellow	SEC 3.
Move to {0,511}, draw to {1023,511}	MOV0. 511. DVA1023. 511.
Enable PAN and observe vectors	EPA (Move joystick - manually)
Position crossing lines at screen center	(Use the joystick)
Disable PAN, enable Joystick	DPA EJC

SVO

Set Vertical Origin				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
SVO	e	101	145	65

SVO

PURPOSE:

The Set Vertical Origin (SVO) command allows the user to set the vertical origin register. Also see VSR command.

FORMAT:

SVO | value

Where:

value (0~511/1023, o-type) is the y coordinate of video memory which is to correspond to the top edge of the displayable screen.

USAGE:

The SVO command is used to change the vertical references of the AED 512/767 display. Video memory is "wrapped around"; that is the part that is not displayable at the top of the screen is displayed starting at the bottom of the screen. Together with the SHO command (see SHO), the user may change the AED 512/767 display coordinate system reference point and view off screen portions of video memory. See also BSO.

NOTES:

1. The default value is 482 or 511 (for 512 line system) for the AED 512, and 574 for the AED 767.

EXAMPLE:

Draw a 100 pixel radius circle and, with the SVD command, divide the circle into two equal halves on the upper and lower portions of the display.

<u>DESCRIPTION</u>	<u>TERMINAL COMMANDS</u>
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Erase Screen	ERS
Move to {100,382}	MOV100.382.
Draw circle with a radius of 100	DCL100. (Note full circle)
Move it vertically	SVD250. (Note semicircle)

Note: The circle moves differently depending on whether you use an AED 512 or 767.

SWM

Set Write Mask				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
SWM	L	76	114	4C

SWM

PURPOSE:

The Set Write Mask (SWM) command allows a subset of the Color Lookup Table (CLT) to be addressed by allowing color number values to be written on a selected subset of the memory planes contained in the terminal hardware configuration. Also see the SRM command.

FORMAT:

SWM | value

Where:

value (0~255, o-type) is an 8-bit byte [B7, B6, ... B1, B0]
and is defined as follows:

if BN = 1, Nth memory plane write enabled
0, Nth memory plane write disabled.

USAGE:

Write masking is used to place an image or bit map in specific memory planes, and then protect those planes from over-writing or erasure. Data subsequently written appears overlaid on the permanent image. Data can be erased without destroying the protected (masked) image. This is accomplished by drawing the image to be protected in a selected group of color numbers (depending upon SWM argument) then protecting them from change or erasure by setting the write mask.

EXAMPLE:

DESCRIPTION -----	TERMINAL COMMANDS -----
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Set write mask to enable all planes for writing (default condition after terminal reset).	SWM255.
Set current color to blue	SEC4.
Simulate a business form with a blue box with corners at {150,400} and {362,300}.	MOV150.400. DFR362.300. SECO MOV160.390. DFR352.310.
Move CAP to start of insert typing.	MOV207.370.
Set write mask to protect blue drawing of box. [11 111 011 base 2 = 251 base 10] [Decimal 255-4=251 blue plane masked]	SWM251.
Set color to red for insert typing.	SEC1.
Leave interpreter, enter mode. [Note interpreter light goes out]	XXX
Type text inside box	AED GRAPHICS
Re-enter INTERPRETER mode	ESC KEY
Erase the screen [Note, blue box is not erased]	HERE IS KEY
Set current color to blue.	SEC4.

(EXAMPLE CONTINUED ON NEXT PAGE)

Move CAP to center of box	MOV256.350.
Draw blue circle, radius 60	DCL60.
<p>Note: That no circle appears on the display because. The write mask is protecting the 100 = 4 base 10 color table address.</p>	
Set color table to define color number 11 base 10 = 1011 as pure blue.	SCT11.1.0.0.255.
Set current color to 11 (blue).	SEC11.
Draw circle, radius 60. [An erasable blue circle is drawn over the box]	DCL60.
Leave interpreter, enter text mode	XXX
Type letters in blue	AED
Re-enter the INTERPRETER mode	ESC KEY
Erase everything except the blue box.	HERE IS KEY
Prepare to erase the protected blue box (form).	SWM255.
Erase	HERE IS KEY
Done	

SZR

Set Zoom Register				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
SZR	E	69	105	45

SZR

PURPOSE:

The Set Zoom Register (SZR) command allows the user to set the X and Y zoom registers independently.

FORMAT:

```
-----  
SZR option 1 option 2  
-----
```

Where:

option 1 and option 2 are the horizontal and vertical zoom values, respectively. (1~16, o-type)

USAGE:

The SZR command is used to zoom simultaneously in both horizontal and vertical directions on the display screen.

NOTES:

The ranges of Option 1 and Option 2 are from {01 to 16}.

EXAMPLE: 512, 767

Horizontally stretch a circle.

DESCRIPTION -----	TERMINAL COMMANDS -----
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Erase Screen	ERS
Move to (255,255)	MOV255.255.
Draw 100 pixel circle	DCL100.
Reset Origin	BSD127.107.
Zoom in X direction only (2 times)	SZR2.1.

VSR

Vertical Scroll Relative				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
VSR	x	120	170	78

VSR

PURPOSE:

The Vertical Scroll Relative (VSR) command allows the user to set the vertical origin register relative to the current vertical origin register contents. Also see SVD command

FORMAT:

```
-----  
VSR Option 1  
-----
```

Where Option 1 is the y coordinate as a signed number.

(-128 ~ 127, o-type)

USAGE:

The VSR command is used to set the vertical origin with respect to its present setting [NEW = OLD + dy].

NOTE:

The y coordinate may range from 0 to 511 on the AED 512 and 0 to 1023 on the AED 767.

EXAMPLE:

Draw a 100 pixel radius circle and with the VSR command, move the circle vertically.

<u>DESCRIPTION</u>	<u>TERMINAL COMMAND</u>
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Erase screen	ERS
Move to {100,200}	MOV100.200.
Draw a circle, r = 100	DCL100.
Scroll down 100 pixels	VSR100.

WDA

----- Write Direct into AOI -----				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
WDA	.	46	56	2E

WDA

PURPOSE:

The Write Direct Into AOI (WDA) command writes image data into the pre-defined area of interest. Also see RDA command.

FORMAT:

```
-----  
WDA ;  
-----
```

USAGE:

The WDA function initiates DMA transfer of words which are 2 bytes each (16-bits). When in 2 pixels/word mode (bit 5 in the SUC command is on (1)), 2 pixel values are defined by each word sent during DMA. If bit 5 in the SUC command is off (0) then 1 pixel value/word is sent.

WHC

----- Write Horizontal sCan/non-ADI ----- FUNCTION CODES -----				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
WHC	u	117	165	75

WHC

PURPOSE:

The Write Horizontal Scan/Non-ADI (WHC) command writes to the terminal the color number for n pixels, beginning with the pixel located at CAP. Also see WHR command.

FORMAT:

```
-----  
WHC : n16 color1 [color2 ... [colorn]]  
-----
```

Where:

n16 (0~65535, o-type) is the number of pixels to be painted

color (0~255, o-type) defines the sequence of colors to be painted; one pixel per color.

[color] = [color1,color2,...colorn] is the sequence.

USAGE:

This command starts with the CAP and paints up to 65,535 (2**16-1) pixels (approximately one fourth of a full 512 x 512 screen). The pixels will be painted one pixel per color for n pixels.

EXAMPLE:

<u>DESCRIPTION</u>	<u>TERMINAL COMMANDS</u>
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Erase Screen	ERS
Set CAP to top left	MOV1.482. (for 512) MOV1.574. (for 767)
Zoom X = Y = 16	SZR16.16.
Write 8 default colors	WHCB.0.1.2.3.4.5.6.7.

WHR

----- Write Horizontal Runs -----				
FUNCTION CODES				

3 CHARS	ASCII	DECIMAL	OCTAL	HEX

WHR	\	92	134	5C

WHR

PURPOSE:

The Write Horizontal Runs (WHR) command writes a sequence of runs. Each run is an instruction to the terminal to paint n pixels of color beginning with the CAP and proceeding horizontally to the right, to the border of the previously established AOI. When reaching the right edge of AOI, the run will automatically return to the opposite edge and jump to the next appropriate scan line. This process will continue within the AOI until a terminating 0 is received. Also see the RHR and WHU commands.

FORMAT:

WHR | (n color) 0

Where:

n (0~255, o-type) is the number of pixels to be written in the following color

color (0~255, o-type) is the color table address for each run.

0 is a terminating zero.

USAGE:

This WHR command is used only in connection with a previously defined Area of Interest (AOI). It minimizes the host overhead in transmitting image data to the terminal by allowing the user to paint contiguous pixels of the same color with only 2 parameters.

NOTES:

1. If the AOI is overflowed by commands to draw excess pixels, the terminal will stop writing but continues to accept bytes until receipt of the terminating zero.
2. The WHR command paints pixels in the direction the AOI was defined. (see DAI)

EXAMPLE:

<u>DESCRIPTION</u>	<u>TERMINAL COMMANDS</u>
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Zoom X = Y = 16	SZR16. 16.
Move Origin (249.260)	BS0249. 260.
Move to lower left of screen	MOV250. 250.
Define Area of Interest to upper right corner	DAI265. 252.
Write series of color runs	WHR8. 1. 7. 2. 1. 4. 7. 4. 2. 3. 8. 3. 8. 5. 5. 6. 3. 1. 0.
	OPPOSITE CORNER OF AOI
Magenta 8 of 5	Cyan 5 of 6 Red 3 of 1
Blue 7 of 4	Yellow 2 of 3 Yellow 8 of 3
Red Cap 8 of 1	Green 7 of 2 Blue 1 of 4

WHS

----- Write Horizontal Scan -----				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
WHS	X	88	130	58

WHS

PURPOSE:

The Write Horizontal Scan (WHS) command is used to fill the entire area of interest by defining each pixel. The area of interest must be defined before this command is issued. (See DAI)

FORMAT:

WHS byte1, byte2, .. byteN

Where;

byte1.. byteN are pixel values to be filled -

$$N=(IX1-IX2+1)*(IY1-IY2+1)$$

where IX1, IX2, IY1, IY2 are assumed as the values used when defining the area of interest as shown:

MOVIX1, IY1, IY1
DAIIX2, IY2

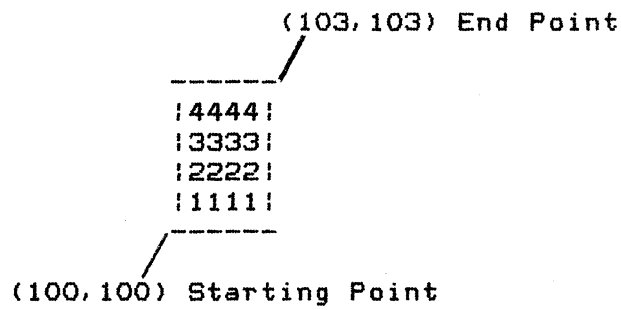
USAGE:

The WHS command starts filling the area of interest from the Current Access Position (CAP) to the corner which is diagonal to the corner defined at the Define Area of Interest (DAI) command. It completes filling at the corner defined by the DAI.

The terminal "waits" for all sequential pixel data to be sent before filling the entire area of interest. Therefore, have a predetermined set area of interest ready to be filled. The CAP is updated on the corner defined by the DAI command when it is completed.

EXAMPLE:

<u>DESCRIPTION</u>	<u>TERMINAL COMMANDS</u>
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Move to {100,100}	MOV100.100.
Define Area of Interest	DAI103.103.
Fill pixels defined	WHS1.1.1.1.2.2.2.2.3.3.3.3.4.4.4.4.



WHU

Write Horizontal rUns alternate

FUNCTION CODES

3 CHARS ASCII DECIMAL OCTAL HEX

WHU s 115 163 73

WHU

PURPOSE:

The Write Horizontal Runs Alternate (WHU) command is similar to WHR in that they both "write horizontal runs" of pixels of the same color. WHU, however, does not utilize an AOI in which to perform. Also see the RHR and WHR commands.

FORMAT:

a. WHU [length color]

b. WHU [255 n color]

c. WHU [length color] [255 n color] (intermixed)

Where:

a. length (1~254, o-type) specifies the run length in pixels

color (0~255, o-type) is the color table address for those pixels

Where:

b. n (1~255, o-type) specifies the number of color table address that follow

color (0~255, o-type) is the color table addresses for n number of consecutive pixels

USAGE:

Combining formats a) and b) will provide the facility to intermix sequences of single pixels of multiple colors with multiple pixels of the same color.

EXAMPLE:

DESCRIPTION -----	TERMINAL COMMANDS -----
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
paint 16 pixels in yellow starting at CAP	WHU16. 3. 0.
paint 16 pixels of yellow followed by 8 pixels of red	WHU16. 3. 8. 1. 0.
paint 16 pixels of yellow, followed by an assortment of 10 pixels consisting of red, white, and blue colors	WHU 16. 3. 255. 10. 1. 7. 1. 4. 7. 1. 4. 7. 1. 1. 0.

WIP

Write Incremental Plot				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
WIP	v	118	166	76

WIP

PURPOSE:

The Write Incremental Plot (WIP) command combines the MOV and WPX functions. The command causes a pixel at the initial CAP to be painted, then the CAP to be moved to an adjacent pixel in the specified direction. This pixel is then painted and the CAP moved to an adjacent pixel in a specified direction. This sequence may be continued to a maximum of 65,536 points. Also see WMP command.

FORMAT:

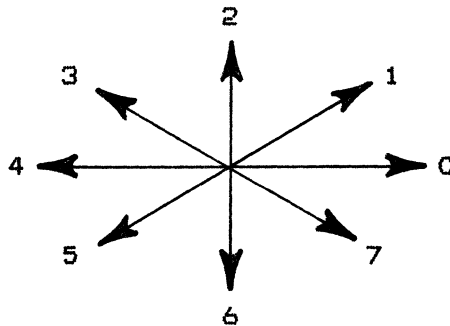
WIP ; n16 byte

Where:

n16 (1~65535, o-type) is a 16 bit (2 byte) operand indicating the total number of pixels to be drawn

byte (0~8, o-type) is the incremental movement encoding, two 3-bit direction codes per byte:

**DIRECTION
CODES**



USAGE:

This command allows the user to write a large number of contiguous but not necessarily linear pixels (as in a data curve) with a minimal number of transmitted bytes.

EXAMPLE:

Use the WIP command to draw the star pattern shown on the following page.

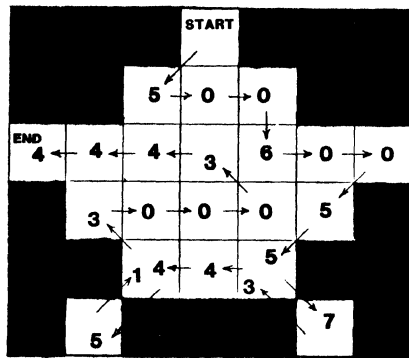
The required terminal command is:

```
RESET RESET LOCAL  
ESC CAPS LOCK
```

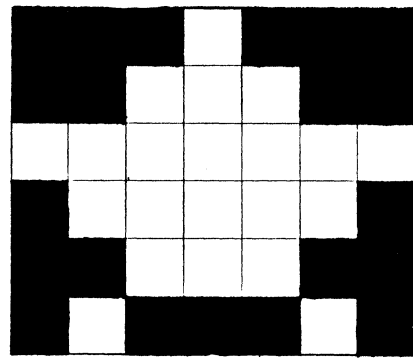
```
SEC7. MOV255. 245.
```

```
WIP24. 40. 6. 0. 45. 59. 36. 41. 24. 0. 28. 36. 0.
```

Activate the cursor and place over the pixel pattern. Depress zoom in key several times and deactivate cursor.



Direction Codes



Star

BYTE CONSTRUCTION									
MSB				LSB					
NOT USED		1 ST			2 ND				
-	-	4	2	1	4	2	1	PACKING	
128	64	32	16	8	4	2	1	ENCODING	
-	-	-	-	-	-	-	-	DECIMAL VALUE	BYTE NUMBER
5,0		1	0	1	0	0	0	40	1
0,6		-	-	0	0	0	1	6	2
0,0		-	-	0	0	0	0	0	3
5,5		-	-	1	0	1	1	45	4
7,3		-	-	1	1	1	0	59	5
4,4		-	-	1	0	0	1	36	6
5,1		-	-	1	0	1	0	41	7
3,0		-	-	0	1	1	0	24	8
0,0		-	-	0	0	0	0	0	9
3,4		-	-	0	1	1	1	28	10
4,4		-	-	1	0	0	1	36	11
0,0		-	-	0	0	0	0	0	12

DIRECTION CODE PAIRS

5,0
0,6
0,0
5,5
7,3
4,4
5,1
3,0
0,0
3,4
4,4
0,0

40	1
6	2
0	3
45	4
59	5
36	6
41	7
24	8
0	9
28	10
36	11
0	12

WMP

Write Multiple Pixels

FUNCTION CODES

3 CHARS ASCII DECIMAL OCTAL HEX

WMP k 107 153 6B

WMP

PURPOSE:

The Write Multiple (isolated) Pixels (WMP) command allows one command with an extended argument, do the work of several pairs of MOV and WPX commands. Also see the WIP command.

FORMAT:

WMP [dx dy]

Where:

dx/dy (-127~128, o-type) are the offsets from the CAP at which the pixel will be painted. CAP is then incremented by dx and dy. Function terminates when dx = dy = 0.

NOTES:

1. Pixels are drawn in current color.

EXAMPLE:

Color red the pixels at the center and corners of a square, 100 pixels on a side and centered at {255,242}. See MVR and WPX for alternate programming.

<u>DESCRIPTION</u>	<u>TERMINAL COMMANDS</u>
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Move CAP to {255,242}, mid-screen.	MOV255.242.
Draw a five pixel image and return CAP to {255,242}.	WMP50.50. ...0.-100... ...0.100.50.-50.0.0.

WPX

Write PiXel				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
WPX	T	84	124	54

WPX

PURPOSE:

The Write Pixel (WPX) command is used to change the color of the pixel at the CAP to the specified color. See WMP command.

FORMAT:

```
-----  
WPX color  
-----
```

Where color is the color table address of the specified color (0 ~ 255, o-type).

USAGE:

To produce an element of an image, the pixel would be changed from the background color (or from a previous foreground image color), while to erase an element the change would be to the background color.

WPX is sometimes used to place visible but unobtrusive location markers on the display, particularly if image construction is being done through the keyboard and zoom or pan are to be used.

NOTES:

Execution of this command does not change the current color (see SEC).

EXAMPLE:

DESCRIPTION	TERMINAL COMMANDS
INITIALIZE TERMINAL	RESET RESET LOCAL ESC CAPS LOCK
SET ENCODING	SEN3DDDN
Move CAP to {255,241}, center of display, and zoom x = y = 16.	MOV255.241. BSO240.255. SZR16.16.
Write pixel to color No. 1, RED	WPX1.
Change pixel to color No. 2, GREEN	WPX2.
Change pixel to color No. 0, BLACK the back- ground color - thereby erasing it.	WPX0.

WRD

Write Raster Direct				
FUNCTION CODES				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
WRD	F	70	106	46

WRD

PURPOSE:

The Write Raster Direct (WRD) command commences host to terminal DVMA data transfer. Host is writing to "the terminal".

FORMAT:

WRD

USAGE:

The WRD command can be issued only from the host device and for transferring pixel values, beginning at CAP and proceeding sequentially through n pixels defined by the word count and address register.

WSF

----- Write Special Font character ----- FUNCTION CODES -----				
3 CHARS	ASCII	DECIMAL	OCTAL	HEX
WSF	8	56	070	38

WSF

PURPOSE:

The Write Special Font (WSF) command writes, at the previously established CAP, the special symbol/character labelled by code, and then moves the CAP by the relative increment dx/1, dy/1; writes the symbol labelled code/2, moves the CAP; etc.

FORMAT:

WSF (code dx dy) 0

Where

code is the character descriptor (1 ~ 255 o type);
dx/dy, is the relative move of the CAP [from
CAP to (CAP + dx,dy)] (-128 ~ 127, o-type).

0 signifies command string terminator.

USAGE:

WSF allows easy access (one keystroke with keyboard entry) to pre-programmed (by DSF command) characters or symbols which are written repeatedly.

XCD

eXit Command DMA				

FUNCTION CODES				

3 CHARS	ASCII	DECIMAL	OCTAL	HEX

XCD	<	60	74	3C

XCD

PURPOSE:

The Exit Command DMA (XCD) is used to exit from command DMA mode which was initiated by the SCD command.

FORMAT:

XCD

APPLICABLE ON AED 767 ONLY

enter eXTenDed command mode

XTD

FUNCTION CODES

XTD

3 CHARS ASCII DECIMAL OCTAL HEX

See
Note 1 + 43 053 2B

PURPOSE:

The Enter Extended Command Mode (XTD) command allows the user to issue the one character command form for certain commands when only 7 bit data is available.

FORMAT:

XTD

USAGE:

The XTD command is used only in the single character encoding mode (see SEN command), to prefix certain commands. These commands are designated in the Notes section of each command description, and include the following:

AAV	BLG	DSP
ELP	ETP	LAT
MAR	RCT	STP
STW		

NOTES:

1. The XTD as a single character, +, command must be issued each time certain single character commands requiring it are used. (AED 767 only).