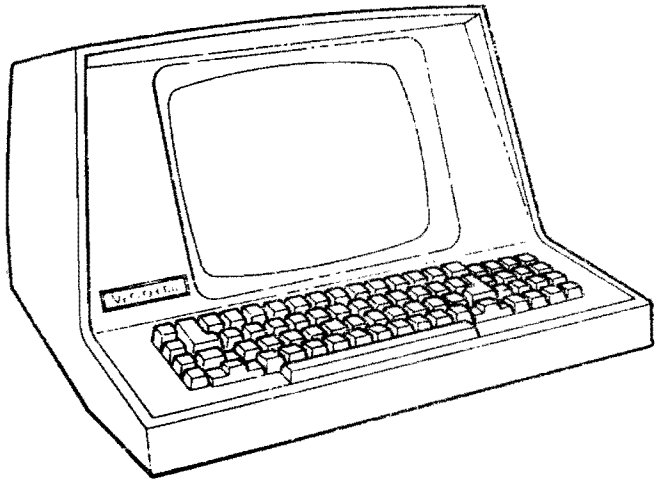
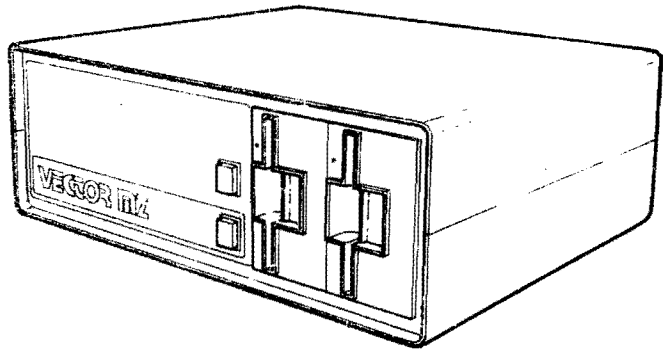
Flashwriter II

Users Guide





©VECTOR GRAPHIC INC.

FLASHWRITER II BOARD

Manual Revision 3
March 29, 1979

Revision Numbers

The revision number and date of release of each page herein appears at the bottom of each page. The revision number and date of release on the Title Page corresponds to that of the page most recently revised.

REPAIR AGREEMENT

The Flashwriter II Board sold hereunder is sold "as is", with all faults and without any warranty, either expressed or implied, including any implied warranty of fitness for intended use or merchantability. However, the above notwithstanding, VECTOR GRAPHIC, INC., will, for a period of ninety (90) days following delivery to customer, repair or replace any Flashwriter II Board that is found to contain defects in materials or workmanship, provided:

1. Such defect in material or workmanship existed at the time the Flashwriter II Board left the VECTOR GRAPHIC, INC., factory;
2. VECTOR GRAPHIC, INC., is given notice of the precise defect claimed within ten (10) days after its discovery;
3. The Flashwriter II Board is promptly returned to VECTOR GRAPHIC, INC., at customer's expense, for examination by VECTOR GRAPHIC, INC., to confirm the alleged defect, and for subsequent repair or replacement if found to be in order.

Repair, replacement or correction of any defects in material or workmanship which are discovered after expiration of the period set forth above will be performed by VECTOR GRAPHIC, INC., at Buyer's expense, provided the Flashwriter II Board is returned, also at Buyer's expense, to VECTOR GRAPHIC, INC., for such repair, replacement or correction. In performing any repair, replacement or correction after expiration of the period set forth above, Buyer will be charged in addition to the cost of parts the then-current VECTOR GRAPHIC, INC., repair rate. At the present time the applicable rate is $35.00 for the first hour, and $18.00 per hour for every hour of work required thereafter. Prior to commencing any repair, replacement or correction of defects in material or workmanship discovered after expiration of the period for no-cost-to-Buyer repairs, VECTOR GRAPHIC, INC., will submit to Buyer a written estimate of the expected charges, and VECTOR GRAPHIC, INC., will not commence repair until such time as the written estimate of charges has been returned by Buyer to VECTOR GRAPHIC, INC., signed by duly authorized representative authorizing VECTOR GRAPHIC, INC., to commence with the repair work involved. VECTOR GRAPHIC, INC., shall have no obligation to repair, replace or correct any Flashwriter II Board until the written estimate has been returned with approval to proceed, and VECTOR GRAPHIC, INC., may at its option also require prepayment of the estimated repair charges prior to commencing work.

Repair Agreement void if the enclosed card is not returned to VECTOR GRAPHIC, INC. within ten (10) days of end consumer purchase.

TABLE OF CONTENTS

I. INTRODUCTON

1.1 Description of the Board

The Flashwriter II Video Board is Vector Graphic's advanced 80 x 24 memory mapped video board. By "80 x 24" we mean that the board displays characters on the screen in 24 rows of 80 characters each. Each character is made up of an 8 x 10 matrix of dots.

The Flashwriter II can be installed in ANY S-100 8080 or Z-80 computer, including existing Vector Graphic computers, in order to convert the existing memory mapped display to 80 x 24, or to make use of memory mapped video for the first time if a serial terminal had been used. It can be used with almost any standard computer video monitor, since it can produce either separate or combined video and sync signals. However, the monitor must have a band width of at least 12 MHz. The board is definitely applicable to the Vector Graphic Mindless Terminal, which requires separate signals. (Note: in this manual, the word "monitor" refers to a video display unit, and the word "Monitor" with an upper-case M refers to a kind of computer program which handles basic housekeeping functions for the computer.)

The Flashwriter II makes use of the most up-to-date ideas in circuit design such as flicker-free updating of the screen, separate sync outputs for video monitors requiring it, on-board user-programmable PROMs storing the character set, on-board socket for a video-driver and/or Monitor, Jump-on-Reset capability, to be used if the on-board video-driver socket is used, the availability of inverted or non-inverted vertical sync signals, the availability of combined or separate video and sync signals, and the capability for reverse video. To make the board adaptable to many different systems, you are free by using jumpers to specify the memory addresses used by the on-board video memory RAM and the on-board video driver.

Further, a parallel keyboard input port is incorporated on the board, with the port numbers changeable by jumper anywhere from 00 and 01 (status and data) up to 0E and 0F. Although the board is shipped jumpered so that it strobes the keyboard data into the input latch on the rising edge of the key-depression strobe, a jumper can be installed to strobe the data on the falling edge, as required by some keyboards. The board can also be jumpered to generate an interrupt when a key is depressed, instead of waiting to be polled.

A particularly useful feature of the board is the ability to accept a user-created character set, which can be either a 128 or 256 character set. If a 256 character set is desired, then the user must sacrifice the use of reverse video. The board is shipped with a 128 character set, on a PROM. This set consists of the 96 standard ASCII characters and 16 special graphics characters which can be used to build graphics images or large characters.

Instructions are provided for creating your own character set and encoding it on 2708 or 2716 EPROMs (the latter for a 256 character set), if desired.

Instructions are given in this manual for writing programs to use the board. In addition, the user can purchase, if not already included in his computer system, the Vector Graphic Extended Systems Monitor with video driver, Version 3 EV-II. Before making this decision, section 3.2.1 should be read. Use of this Monitor will eliminate some or all of the assembly language programming necessary to use the board. The EVIOS program, a sophisticated video driver used with the Flashwriter I board, canNOT be used with the Flashwriter II board, but Version 3 of the Monitor makes it largely unnecessary. For user's familiar with Flashwriter I, this is partly due to the fact that Flashwriter II does not have a separate memory block containing character attributes. In Flashwriter II, there are no special vertical and horizontal lines outside of those in the graphics characters, there is no reduced intensity, and the graphics characters are produced in response to ASCII codes 00 to 1F and 80 to 9F.

The Flashwriter II board is a major addition to the Vector Graphic product line. Vector Graphic has now applied its well known excellence in video display quality and product flexibility to the creation of an advanced 80 x 24 video display board.

1.2 Description of the Manual

This manual provides a discussion of the theory of operation of the Flashwriter II, and a User's Guide describing 1) when and how to modify the board's electronics and PROM's, 2) how to program for the board, how to adjust the TV monitor, and how to connect the board to keyboard and video. Since the board is not sold as a kit, assembly information and parts list are not included.

II. THEORY OF OPERATION

A block diagram of the video display module is shown on page 4-5. Each of the blocks is comprised of several integrated circuits as shown on the schematic diagram on page 4-7.

2.1 Keyboard Port

Starting with the keyboard port, U46 is an 8 bit latch which internally stores the data from the keyboard coming through J1 when the STB (pin 11) input goes low. A jumper option allows the correct strobe polarity to be selected. U46 contains a service request flip flop which is set by the same strobe edge that latches the data, causing $\overline{INT}$ to go low (pin 23). This signal can optionally be connected to $\overline{PINT}$ on the bus (pin 73) for interrupt driven keyboard input, or can be tested by accessing the status port. Two adjacent ports are always occupied by the status and data with the status being the lower one. The port address is decoded by U39 and the associated sections of U27 and U38. A jumper allows the selection of port addresses from 0 and 1 to E and F. (Each pair of ports refers to status and data, respectively.) Signals INPS and INPD enable tristate bus drivers U45 and U21 to gate the $\overline{INT}$ signal and data onto the bus at the appropriate time. Both inverted and non-inverted status are available, for keyboard drivers using different conventions. The vertical blanking signal is also available, from the status port, for use as a 60Hz clock signal, or for synchronizing data transfers with the vertical retrace.

2.2 Horizontal Sync Circuitry

The timing for the characters, horizontal blanking and horizontal sync pulse is provided by U15, U29, U3 and U1. A crystal oscillator at 14.318 MHz provides the clock for all the signals. This is the frequency at which the individual dots making up the characters are displayed. U15 divides this clock by 8 to generate a character clock output every time a new character is to be displayed. This signal is further divided by U29 and U3 to generate a horizontal period of 63.69 micro-seconds. The outputs H0-H6 are binary outputs representing the 80 character positions per line. H7 goes high at the end of the displayed line of characters, and is used as the horizontal blanking signal. At the count of 207, decoded by U2 pin 8, the counter string is preset to the value of 94 and starts counting over again. This signal also triggers U1, a dual one-shot to generate a horizontal sync pulse. The delay of this pulse can be varied by the horizontal position potentiometer to allow centering the display on the TV screen.

2.3 Vertical Sync Circuitry

The vertical sync counters U6, U4 and U5 are clocked by the horizontal sync pulse from U1 pin 5 to produce a count from 0 to 261. Each row of characters occupies 10 scan lines. U6 is a decade counter, so it generates a terminal count (TC) every 10 lines. The displayed characters occupy 240 lines, and U19 pin 12 goes low at the count of 240 to blank the display. A vertical sync pulse is generated by a section of U5 and U20 pin 1 which is 2 lines long, from 240 to 242. This is slightly shorter than a standard TV sync signal, and produces a minimum disturbance to the horizontal sync of the TV. The video output remains blanked until the counters are preset to 0 at the count of 261 for a total of 262 scan lines.

2.4 Memory Address Multiplexer

The on board memory is multiplexed between the CPU and the sync circuitry. This is done using tristate drivers U40, U41, U28 and the tristate outputs of U16. Thus when the CPU addresses memory, the address bus signals drive the memory address inputs, and while the characters are displayed on the screen, the address inputs are generated by the sync circuitry. One complication is that since 80 is not a binary power, there is not a convenient separation of horizontal and vertical address inputs. 80 is divisible by 16, so the least significant 4 bits from U29 are applied directly to the memory address inputs. The remaining 3 horizontal counter outputs H4-H6 and the vertical sync signals V0-V4 are mapped into 7 bits using a 256 x 8 ROM U16. This effectively maps the displayed characters in a linear fashion into the 2K of address space occupied by the memory. Since 80 X 24 = 1920, there are 128 locations in RAM that are not displayed on the screen.

If the memory is addressed by the CPU, it will not be able to generate the proper output to be displayed on the screen. This would cause undesirable glitches, or flashes of light on the screen as the display was being written into. To prevent this, access is inhibited by the CPU except during the horizontal retrace interval. U12 provides the necessary arbitration and pulls PRDY (pin 72) low to put the CPU in a wait state until it can access the memory. Two sections of U12 provide delays to ensure that control is transferred properly. U12 pin 13 provides a delay to ensure that the CPU has finished its current memory access cycle, and U12 provides a similar delay to ensure that the memory access time is satisfied before the CPU proceeds. The data bus is buffered by U44 and U33.

2.5 Memory Latch

The combined access time of the memory and the character generator ROM exceeds the character period of 558 ns. Thus the data is "pipelined" using U34 and U35 to latch the memory data. This

provides 558 ns access time for both the RAM and character generator. The output of U34, U35 is delayed by one character clock period from the sync counter outputs, while the data strobed into the dot shift register U8 is delayed two character clocks. To compensate for this delay, it is necessary to delay the horizontal blanking signal also. This is done using sections of U35, and the output of pin 12 is H Blank DLD, the delayed horizontal blanking signal.

2.6 Character Generator and Shift Register

The dot patterns for each character are generated by U22 and U23. The 8 bit output from these user programmable EPROMs is supplied in parallel to U8 which shifts the dots out at the high dot clock (14.318 MHz) rate. The addressing of U22, 23 is arranged so that there are several options as far as the ROMs are concerned. U22 generates the top 8 lines of each character cell. If only upper case characters are used with no descenders, this is the only ROM required. For the descenders of lower case characters and graphic symbols, a second ROM is required (U23) which generates the bottom two lines of each character cell. If in addition, a full 256 characters are used, U22 is replaced with a TMS 2716. For the 128 character ASCII character set, the most significant memory bit is not required. This is normally jumpered to U7 pin 12 to control the reverse video. If the 256 character set is used, this bit is connected to U22 pin 20 to select the upper half of the ROM. More information on this subject is provided in the User's Guide, Section III of this manual.

2.7 Video Combiner

The horizontal and vertical sync signals are available at J2 pin 4,5 for monitors such as the Ball Brothers TV 120 used in the Vector Graphic Mindless Terminal which require separate sync and video. U19 pin 8 is the combined video and blanking signal which is available at J2 pin 3 with the proper polarity for the Mindless Terminal. The sync signals and video are combined in the circuitry associated with U10, and are available at J2 pin 1. The horizontal and vertical sync are first combined in U7 and then summed with the video using the open collector outputs of U10 and the resistive network. The resistor values have been chosen to give the proper sync and video amplitudes and to provide a 75 ohm source impedance to drive a terminated video cable. Very good video rise and fall times are obtained with this circuit. In order to compensate for the limited bandwidth of most TV monitors, some high frequency preemphasis is provided by the 470 pf capacitor shunting the 100 ohm output resistor. If the video display is not satisfactory, it may be improved by changing this value.

2.8 Monitor ROM Circuit

A socket is provided for a 2708/2716 ROM to be used as a monitor and/or video driver. U26 selects both the address for the video memory RAM and also the ROM with jumper options every 2K in the upper 16K of memory. A Jump on Reset flip flop consisting of U9 sections has two functions: 1) to disable RAM after the system is powered up or reset; 2) to enable the on-board Monitor ROM at address 0. If the first three instructions of the ROM are JMP XX03, where XX00 is the normal ROM address, then when the system is powered on or the reset key is depressed, these instructions will be executed, causing the CPU to continue executing with the 4th byte of the ROM. After the jump takes place, the circuitry automatically resets the flip flop and restores normal operation of the RAM. Circuitry is also provided to generate MWRITE, a signal produced in some computers by the front panel. This is not required in Vector Graphic equipment as it is generated by the Z-80 CPU board.

[illegible]

3.1 Modifications [illegible]

The sub-sections [illegible] of the possible modifications [illegible] Board, both to [illegible] which cannot [illegible] board. In each [illegible] of the board is used [illegible] might make the [illegible] carrying [illegible] This section [illegible] determine what [illegible] constitutes the [illegible] board. In order to find the location [illegible] within each area, refer to the diagram [illegible]

3.1.1 Creating a [illegible]

The Flashwriter II [illegible] standard 96 character ASCII character set [illegible] characters. You will find a diagram of [illegible] The graphics characters can be used [illegible] programming to create a large variety of [illegible] screen. These 128 characters are those [illegible] on the video screen when the corresponding [illegible] the on-board RAM. The code for each character [illegible] the lower 7 bits of each stored byte. The eighth [illegible] converts the character into reverse video (black on [illegible]) Otherwise it is normal video. Reverse [illegible] graphics characters are found on page 4-2.

The characters are stored [illegible] U22 and U23. Each character consists of an [illegible] as illustrated on page 4-1. PROM U22 stores the first [illegible] (lines 0 to 7) and U23 stores the last 2 lines [illegible]. The lowest addresses on U22 contain the first [illegible] 128 characters, then come the second lines of [illegible] and so on, up to the eighth line of all 128 [illegible] the characters with the [illegible]

If you wish to replace these characters, you need only create a new pair of PROM's, using the arrangement described above. For a nominal fee, Vector Graphic will do this for you. You must provide Vector Graphic with a drawing of each character, darkening the appropriate cells in an 8 by 10 matrix. Use the blank character form on page 4-3. Note that if you limit yourself to only upper-case characters, not using the bottom two lines in each character, you will not need the U23 PROM. To order custom character PROM's from Vector Graphic, please contact the company directly to make arrangements.

You can also create a full 256 character set. To do this, you use a

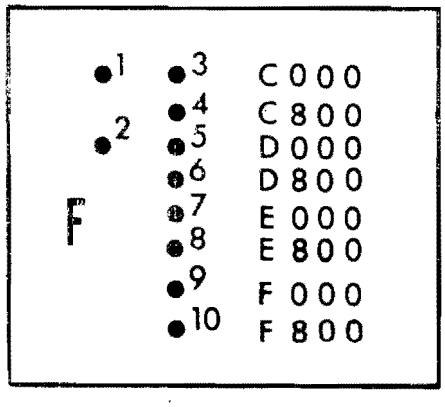
TMS 2716 2K x 8 PROM or equivalent for U22, and 2708 for U23. The data is organized in the same way as with a 128 character set, except that 256, instead of 128, consecutive addresses contain the first lines of all the characters, and so on for each line in the character matrices. With a 256 character set, you sacrifice the use of reverse video, because all 8 bits of each byte are used to designate the character. Vector Graphic will also create the PROM's for you for a 256 character set, for an appropriate fee. Use both of the blank character forms found on pages 4-3 and 4-4 when sending your character set to Vector Graphic, putting the first 128 characters on the first page, and the second 128 characters on the second page.

The board is shipped jumpered for a 128 character set. Jumpers must be changed as follows for a 256 character set.

	128 character set	256 character set
Area I	2 - 1	Cut 2 - 1
Area B	2 - 4 3 - 1	2 - 1 3 - 4 Cut 2 - 4 Cut 3 - 1
Area E	1 - 3 2 - 5	1 - 3 Cut 2 - 5 2 - 4

3.1.2 Changing the Address of the Video Memory

2K of RAM are provided on-board for storage of the current screen image. Since the screen contains only 1920 character locations, the last 128 bytes of this RAM is available for any other purpose. The board is shipped with this RAM addressed at D000. Since the Vector Graphic Extended Systems Monitor assumes this location, do not change it if you are using the video driver in this Monitor, unless you are willing to modify the Monitor on PROM. If you do want to change the address of video memory, a jumper must be changed in Area F. The jumper which determines the address of the RAM goes from pad 2 to one of the 8 pads below it. Each of the 8 pads corresponds to one address, as indicated in the diagram below. Cut the existing jumper and install a new one as required.



3.1.3 Putting a Monitor PROM on the Board

The Flashwriter II board is not shipped with a Monitor PROM on the board. However, socket U42 is available for either a 1K 2708 or a 2K 2716 PROM holding a Monitor and/or video driver. This program can be one of the Vector Graphic Monitors (see section 3.2.1) or one you have written yourself (see section 3.2.2.) (Note that in Vector Graphic computers the Monitor PROM resides on the 12K PROM/RAM board.) If you do install a Monitor PROM on the Flashwriter board in U42, it will function if the board is properly jumpered. Refer to sections 3.1.4, 3.1.5, and 3.1.6.

3.1.4 Address of the Monitor PROM on the Board

If you choose to use the socket provided on the board for a Monitor and/or Video Driver PROM, you must put a jumper in Area F to specify the address of this PROM. The jumper goes from pad 1 to one of the 8 pads below it. Each of the 8 pads corresponds to one address, as indicated in the diagram in section 3.1.2. Obviously, you cannot use the same block of memory that is used for the on-board RAM.

3.1.5 Specifying whether Monitor PROM on Board is 2708 or 2716

If you choose to use the socket provided on the board for a Monitor and/or video driver PROM, the chip used can be either a 1K or 2K chip, i.e. a 2708 or 2716 respectively. The board is shipped to accept a 1K PROM, as determined by the jumpers in Area D. In order to use a 2716 PROM, cut the jumpers from 1 - 4 and from 2 - 3 and replace with a jumper from 1 - 3 and a jumper from 4 to Area M. (There is only one pad in Area M.)

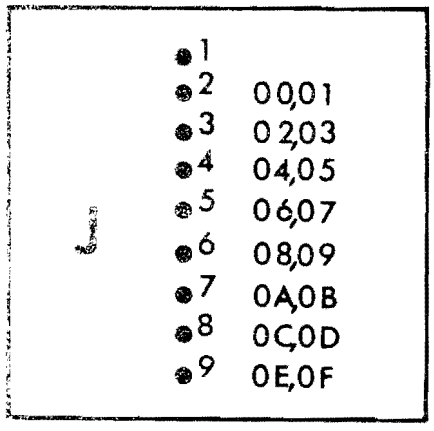
3.1.6 Enabling Jump on Reset

At this time, the Flashwriter II board is not shipped with a Monitor PROM, and therefore is not responsible for Jump-on-Reset. If you install a PROM on the board, you will want to enable Jump-on-Reset. If this is enabled, when the operator powers on the system or depresses the front panel reset switch, the CPU will automatically read the first 3 bytes of the Monitor PROM on the board. Thus, the first three bytes of the Monitor PROM must be a jump to some other address in memory. Usually this address is simply the next address on the same PROM, namely the beginning address of the PROM plus 3.

In order to enable Jump on Reset, install a jumper in Area C from 1 - 2, and in Area K from 1 - 2. If you are writing your own Monitor PROM, then the program which begins at the 4th byte of the PROM must be an appropriate response to the reset.

3.1.7 Changing the Keyboard Port Address

Normally the keyboard is accessed through ports 00 and 01 (status and data, respectively). You can change this to any consecutive pair of ports up to 0E and 0F. This is done by changing the jumper in Area J. Cut the existing jumper. Then, install a jumper from pad 1 to the pad corresponding to the desired port address, as illustrated below.



3.1.8 Changing the Polarity of Vertical Sync Signal

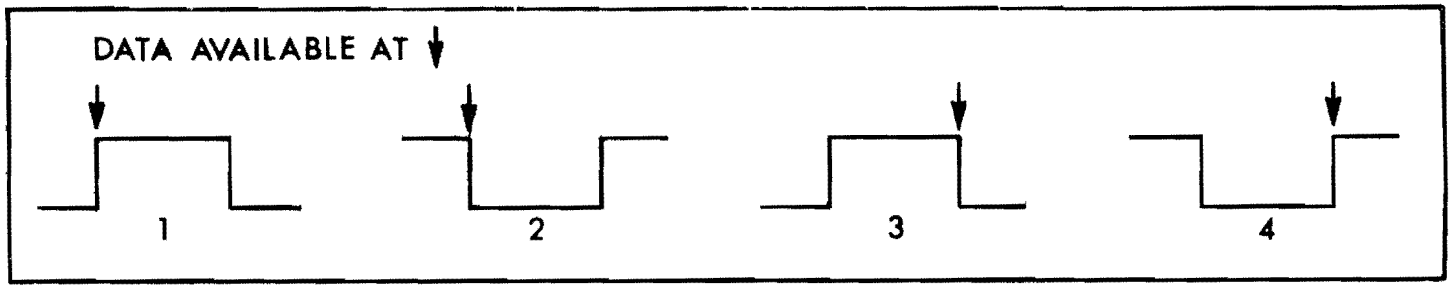
The board is shipped to output an inverted vertical sync signal, as required by the Vector Graphic Mindless Terminal. If you are using a video monitor which requires a non-inverted vertical sync signal, then cut the jumper in Area A from 1 - 2, and install a jumper from 1 - 3.

3.1.9 Obtaining MWRITE

The composite S-100 signal MWRITE is produced by the Vector Graphic Z-80 CPU board. In other computers, it is produced by the front panel. If your computer does not generate MWRITE, then you will need it in order to write to memory. The Flashwriter II board will generate it if you place a jumper in area G, from 1 - 2.

3.1.10 Changing the Polarity of the Keyboard Strobe

There are 4 common types of strobes generated by keyboards to indicate that a key has been depressed, as illustrated below:



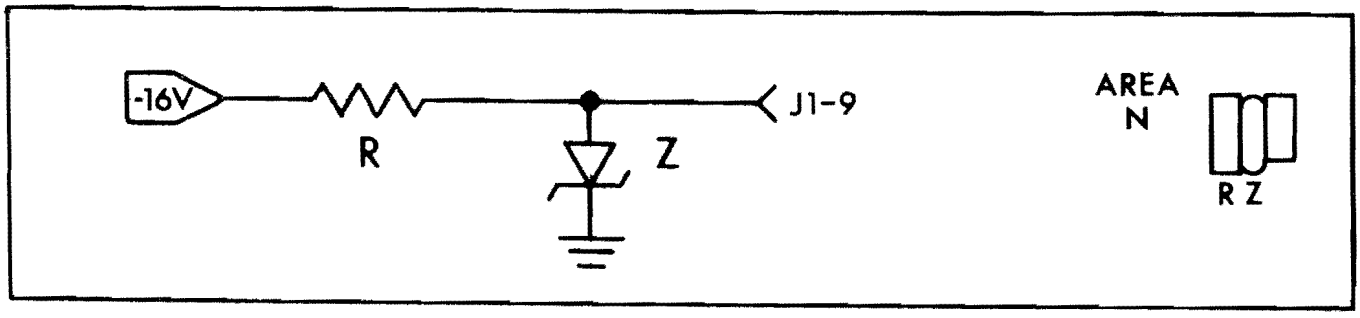
The first type is the one used by the keyboard on the Vector Graphic Mindless Terminal. Both it and the fourth type require that the key data be strobed into the Flashwriter keyboard latch on the rising edge of the strobe. The Flashwriter II board is shipped to strobe the data into the latch on the rising edge. On the other hand, if your keyboard generates the second or third type of strobe, cut the jumper in Area H.

3.1.11 Using Interrupt Driven Software

If you want the keyboard to generate an interrupt whenever a key is depressed, put a jumper in Area L from 1 - 2. If this jumper is not in place, then keyboard data can only be obtained by polling the status port. (See section 3.2).

3.1.12 Supplying a Keyboard with a Negative Voltage Power Supply

If you have a keyboard which requires a negative voltage power supply in addition to the +5V, and you are technically oriented, then there are pads in Area N of the board to install a zener regulated power supply. The circuit diagram on the left, below, will be completed if you insert the correct resistor and zener in Area N, as shown on the right, below.



The zener voltage will depend on the keybard requirements if it is needed at all, and the resister should be selected to bias the zener with at least 10 mA of current in addition to the current required by the keyboard. For example, with a keyboard requiring 10 mA of current at -6V, the zener could be a IN752A (5.6V) and the resistor could be 10/.02 = 500 ohms (470 nominal). The zener power dissipation would be 60 mw and the resistor dissipation would be 200

mw. Use a 1/2 watt resistor to allow for higher supply voltages.

3.1.13 Using a Spare Key on the Keyboard for Reset

If your keyboard has an extra key not connected to the keyboard encoder logic, it can be used to reset the system, as an alternate to the reset key on the front panel. This is not possible with the Vector Graphic Mindless Terminal. Simply connect the key so that when it is depressed it grounds pin 11 of J1, the keyboard socket. See section 3.4.3 for a diagram of J1.

This pin is connected to pin 75 on the bus (PRESET). When the line is grounded by depressing the key, the jump-on-reset circuit on this or another board will enable the Monitor PROM on the same board.

3.1.14 Accessing the 60 Hz Vertical Blanking Signal

Bit 5 of the input status port is low during vertical retrace.

3.2 Programming for the Flashwriter II Board

3.2.1 Vector Graphic Extended Systems Monitor 3 EV-II

By far the most convenient way to use the Flashwriter II board is through the use of the Vector Graphic Extended Systems Monitor, Version 3, Option EV-II or CV-II. This program comes on a PROM to be installed in position 8 on the Vector Graphic 12K PROM/RAM board. The Monitor is NOT automatically included with any order of the Flashwriter II board, and therefore must be ordered as a separate item. If you are upgrading a system from a Flashwriter I board, the Extended System Monitor used for that board will not work with Flashwriter II. You must order Monitor 3.

In contrast with earlier Vector Graphic Monitors, the video driver in Monitor Version 3 allows you to write anywhere on the screen, either by moving the cursor from the keyboard or entering X,Y coordinates from an assembly language or BASIC program. (BASIC must have POKE, as does M.BASIC provided with Vector Graphic Systems.) It also allows you to toggle reverse video from program or keyboard, and offers several less significant additional features. Both Monitor Version 3 and earlier versions contain keyboard input routines. Option EV-II interfaces to parallel keyboards, while Option CV-II interfaces to serial keyboards as on printing

terminals.

In addition to the video driver, and keyboard input routines, the Version 3 Monitor offers a wide range of useful utility programs, including ASCII Dump, Hex Dump, Jump to Micropolis Bootstrap Loader (D800), Compare Blocks of Memory, Jump to Extension PROM (C400 - the start of MZOS or a user written PROM), Find Two Bytes, Go To and Execute, Input from a Port, Jump to Loaded DOS (jumps to the warmstart location of MZOS or MDOS, whichever is running), Jump to 0000, Move Memory Block, Non-destructive Memory Test, Output to Port, Accessing Program Memory (for displaying and changing consecutive addresses), Compute Checksum, Jump to DC00, Search for Single Byte, Test Memory, Jump to 2A00, Wide Screen ASCII Dump, Exchange Memory Blocks, Keyboard Echo, and Zero or Fill Memory. These routines are accessed when the Monitor Executive routine is running, as indicated by the Monitor prompt *.

Version 3 of the Monitor contains several methods of displaying characters. Although the documentation to the Monitor describes these, a review here is useful. The basic technique is to call the video driver (CALL C006) with the desired character in the A register. This is sometimes called "sending" a character to the driver. This is the method used by all Vector Graphic operating system software such as MDOS and MZOS to display characters. By itself, this can only be used to display normal alphanumeric characters, those with codes between 20 and 7F (Hex). (The driver converts the codes higher than 7F to the corresponding code from 00 to 7F, by changing the 8th bit from 1 to 0.) Then, only the codes between 20 and 7F are displayed. Any code between 00 and 1F is interpreted as a command rather than a character, or if not a valid command, then ignored.

How then are graphics characters displayed? To get around the above resrictions in order to print characters outside the range 20 to 7F, first put the character you want to print in the B register, then put Hex 05 in the A register, and then call the video driver at C006. If the character is from 80 to FF, then it will be displayed in reverse video if your board is jumpered for reverse video as shipped. If not jumpered for reverse video and you are using a 256 character set, then codes 80 to FF will produce whatever characters are specified in the character generator PROM (see section 3.1.1.) If the character is from 00 to 1F, the corresponding graphics character will be displayed, or whatever other character is stored in the character PROM if you have created your own character set.

cancel, send another Control-T.

3.2.2 Displaying Characters Without Using the Vector Graphic Monitor Video Driver

This section is of concern if you are bypassing the Vector Graphic Extended Systems Monitor video driver, or writing your own video driver.

To display a character somewhere on the screen, simply write the corresponding ASCII code into the appropriate RAM location, using the RAM on the Flashwriter II board. Unless you have changed the location of this RAM as described in section 3.1.2, it is the 2K block from D000 to D7FF. Since the screen is 80 x 24, the last 128 bytes of this block are not used. The first location of this RAM corresponds to the upper left-hand corner of the screen, and successive locations move across the screen from left to right, going to the left edge of the next line down at the end of each line.

If the board is used as shipped, the characters which will be produced by each ASCII code are shown on the diagrams on pages 4-1 and 4-2. This includes special graphics characters which you can use to build graphic images such as pictures or large letters. Note that the characters from 80 to 9F are reverse video versions of those from 00 to 1F. Similarily, if you display a character having a code from A0 to FF, it will be the reverse video version of normal alphanumeric characters from 20 to 7F. (This assumes that the board is jumpered for reverse video, as shipped.) As explained in 3.1.1, you may create your own characters to replace those supplied with the board.

If you are writing your own Monitor program, including a video driver, you can if you choose install this PROM on the Flashwriter Board. The socket is labelled U42 on the board. If you do this, you should refer to sections 3.1.3, 3.1.4, and 3.1.5 in order to make appropriate hardware modifications if necessary.

3.2.3 Producing Reverse Video

This section specifically covers reverse video, even though it has been discussed in preceding sections. Reverse video refers to displaying a character in black, on a white background. This section is only relevent if the Flashwriter Board is jumpered for reverse video, as shipped from Vector Graphic, rather than for a 256 character set. (See section 3.1.1.)

The easiest way to cause reverse video is to go the Monitor Executive routine (if you have the Vector Graphic Monitor Version 3) by depressing ESC on the keyboard. Then depress Control-T (CTRL and

T keys simultaneously). Any characters displayed after that will be in reverse video, until another Control-T is depressed.

Any character stored in the Video Memory RAM will be displayed as a reverse video character if the 8th bit of the character is a 1. This corresponds to Hex codes 80 to FF. If the 8th bit is 0, i.e. codes 00 to 7F, video will be normal. In other words, if you display a character with a code between 80 and FF, it will be the reverse video version of the corresponding character between 00 and 7F. The chart on page 4-1 shows characters corresponding to 00 to 7F (Hex). On page 4-2 a number of additional graphics characters are shown having codes from 80 to 9F (Hex). Notice that the graphics characters from 80 to 9F are reverse video versions of those from 00 to 1F. In the same way, if you store in video RAM the codes beyond 9F, that is A0 to FF, they will be displayed as the reverse video versions of the normal ASCII characters, having codes 20 to 7F.

There are three ways to store the reverse video characters in video memory. First, you can create the appropriate 8 bit code in a program and then store it in video RAM. For the second and third methods you must have the Vector Graphic Monitor Version 3.

As the second method, you can send the character Control-T (14 Hex) to the video driver. This is how the method given in the first paragraph of this section works. Control-T toggles reverse video, so that any characters sent after it, assuming they are in the displayable range, i.e. between 20 and 7F, will be displayed in reverse video (i.e. the driver will set the 8th bit.) The Control-T itself will not be displayed.

(Sending a character to the video driver means putting the code in the A register and calling C006. The Monitor Executive routine, and M.BASIC and MDOS do this automatically when displaying a character on the screen, so that all you have to do is cause the character to be displayed in any way available. For example, since the Monitor Executive echoes characters as they are entered, just depress Control-T on the keyboard after the Monitor prompt * in order to toggle reverse video. In contrast, the MZOS executive will pass Control-T (and all other Control characters) only if it is the first character of a line. The CP/M executive will not pass it at all, so that you cannot send a control-T from the keyboard under the CP/M executive.

Third, you can send the character Control-E (05 Hex) to the video driver. Following this, you can put any character code in register B and call C006, and the code will be displayed. If it is in the range 80 to FF, then it will be reverse video because the 8th bit is set in these codes. Refer to the Monitor documentation for more discussion of Monitor commands.

3.2.4 Determining Whether a Character is in the Keyboard Buffer

This section is of concern if you are not using the keyboard input routine in the Vector Graphic Monitor. Bit 6 of the status port (usually port 00, unless the board is jumpered as described in section 3.1.6) is high and bit 0 is low when a character is available in the data port. You can test either one of these bits. When the test is successful, simply input the data from the data port (data port = status port + 1).

3.2.5 Accessing the Vertical Blanking Signal

For programmers interested in accessing the vertical blanking signal, bit 5 of the status port is low during vertical retrace and high at all other times.

3.3 Adjusting Your TV Monitor

If your monitor is the Mindless Terminal from Vector Graphic, then refer to the handbook for the Mindless Terminal instead of this section.

The FLASHWRITER is designed to utilize every line of the TV raster (picture). Therefore, some adjustment of the TV monitor is usually required to make all of the characters visible on the screen. Other adjustments may also be necessary. Most monitors have the following controls, either at the rear of the set, or as in the case of the Hitachi, inside the back panel. Often times an insulated screw driver is necessary to turn the controls.

1. HEIGHT - controls the amplitude of the vertical deflection.
2. VERTICAL LINEARITY - controls the line spacing usually at the top of the picture.
3. WIDTH - controls the amplitude of the horizontal deflection.
4. CENTERING - controls the positioning of the raster on the tube.
5. HORIZONTAL HOLD - sets the frequency of the horizontal oscillator.
6. VERTICAL HOLD - sets the frequency of the vertical oscillator.
7. BRIGHTNESS - sets the background picture brightness.
8. CONTRAST - sets the video amplifier gain.

Begin the adjustment of the set with the HEIGHT and VERTICAL linearity. Fill the screen with a pattern of characters. (If you are using the Vector Graphic Extended Systems Monitor, then enter Z

D000 D7FF 30 following the Monitor prompt. The missing character in the bottom line is normal - it is the cursor location.) Adjust the HEIGHT control, which is usually at the rear of the set, until all the characters fit on the screen with an adequate margin on top and bottom. Notice whether the top line of characters is exactly the same height as the bottom line. If it is not, adjust the VERTICAL LINEARITY control until it is. These controls tend to interact to some degree, so several adjustments back and forth may be necessary until a satisfactory adjustment is obtained. Do not be concerned if the display is not exactly centered from top to bottom. The HORIZONTAL HOLD should now be adjusted so that the control is in the center of the range over which the display is in sync, i.e., no characters are misshapen. For some types of displays, this may be over the whole range. At this point, the horizontal position control on the Flashwriter Board should be adjusted so that the rows of characters are centered on the screen horizontally. This control is the small black potentiometer on the upper left hand corner of the board.

Hopefully, at this point you have satisfactorily adjusted your display. If the characters extend off the sides of the display, it will be necessary to adjust the WIDTH control. Some newer sets do not have WIDTH controls. If you are electronically oriented, you can handle this as follows: You can install a choke in series with the horizontal deflection yoke. Local radio-TV stores may be able to supply a width choke if this is necessary, or it is possible to wind one on a small powdered iron form on a cut-and-try basis. Another possibility is to slightly increase the high voltage if the set has this adjustment.

After this, if the display is not exactly centered, it may be desirable to adjust the centering rings on the deflection yoke. On the rear of the deflection yoke are two metal rings with tabs protruding from them. These rings are magnetized, and by rotating them independently, the display can be shifted in any direction up to 1/2 inch or so.

The adjustment of the BRIGHTNESS and CONTRAST should be so that the background is just barely blacked out or slightly grey, while the characters are just bright enough. Too much contrast will result in excessive overshoot on the left edges of the characters, or "hot spots" in the characters.

3.4 Connecting the Board to the Video Display and the Keyboard

This section is only relevent if the board is purchased as a separate item, not already integrated into a computer at the factory.

The 6-pin molex connector, called J2, jutting from the top left corner of the board is used to connect the board to the video

display. The pins are numberd 1 to 6 from the left. Pin 1 is composite video, pin 2 is ground, pin 3 is TTL video, pin 4 is horizontal sync, and pin 5 is vertical sync. Accompanying the board in the same shipping container, you will find a small enevelope containing one 6-socket molex connector, and 6 small pins that insert into it. Solder the wires of a cable of your choice onto as many pins as you need. Then, insert these pins into the sockets corresponding to the pins on J2 which you require, as described in sections 3.4.1 and 3.4.2.

As an alternate to using the enclosed 6-socket connector and assembling your own cable, Vector Graphic supplies two cables factory assembled, which you can use if one or the other meets your needs. They are described in sections 3.4.1 and 3.4.2.

3.4.1 Connecting a Video Display Requiring Separate Sync and Video

Access pins 2, 3, 4, and 5 of J2.

The cable supplied with the Vector Graphic Mindless Terminal comes with an appropriate socket attached. If you have ordered the Vector Graphic Mindless Terminal, then this cable will be enclosed with it. Use of this cable is described in the Mindless Terminal manual. You may discard the socket and insertable pins accompanying the Flashwriter II board.

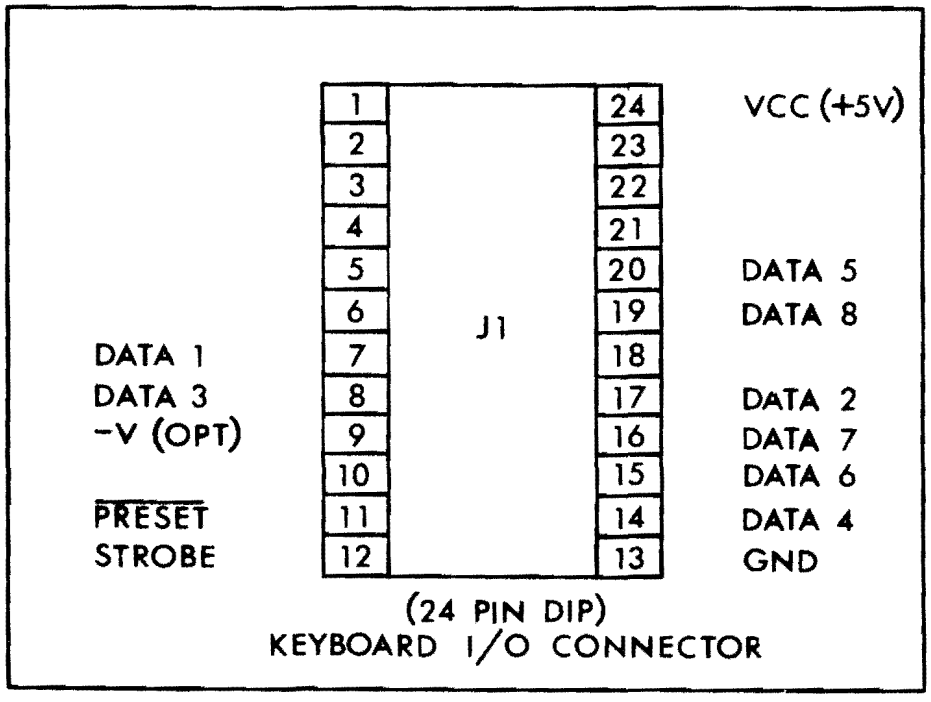
3.4.2 Connecting a Video Display Requring Combined Sync and Video

Access pins 1 and 2 of J2. If you are using a monitor having a coaxial cable, you will probably want to assemble a cable having a compatable socket at one end, and install it in the rear of your computer, with the other end connected to J2 of the Flashwriter Board.

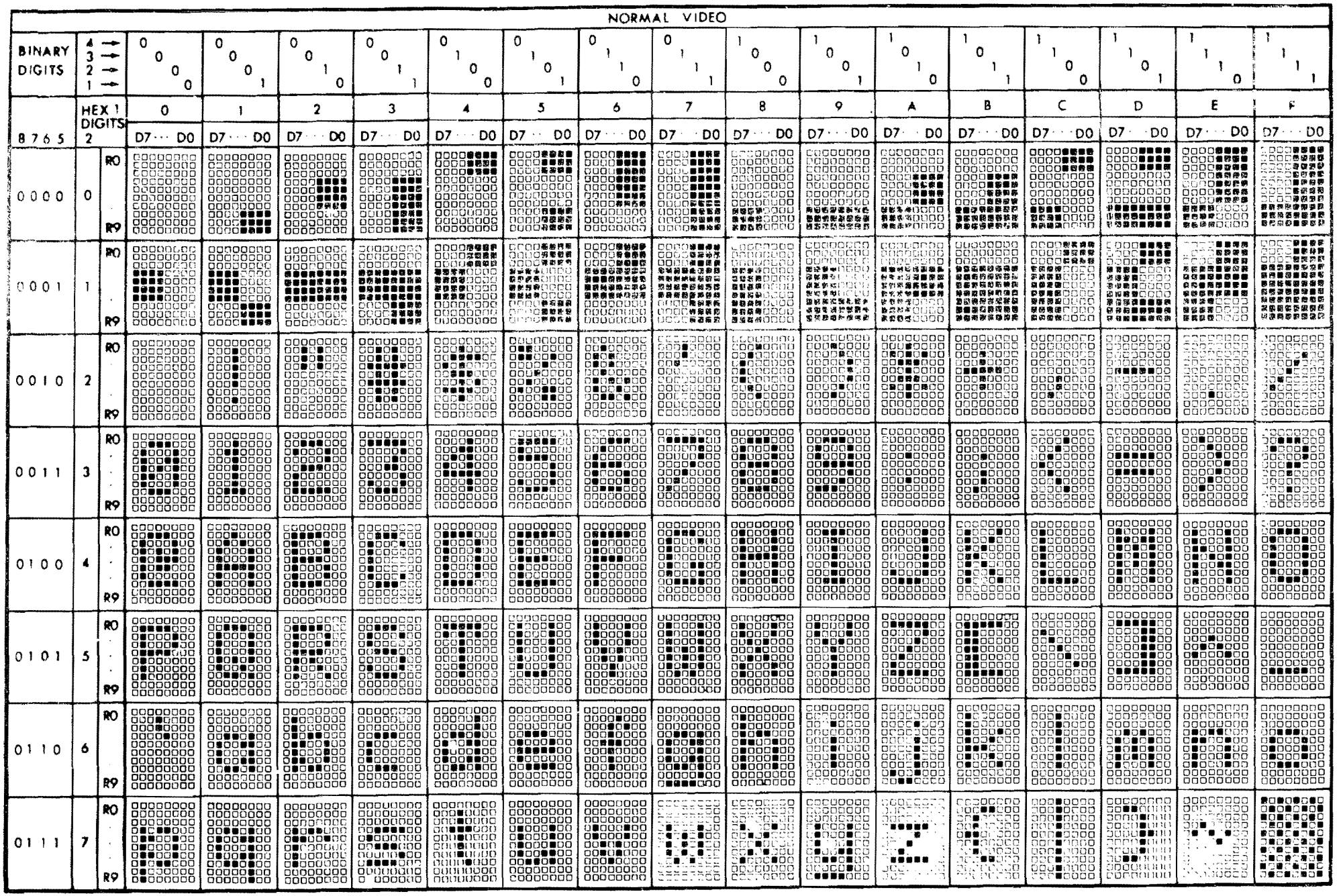
You can order from Vector Graphic a cable equiped with a 2-socket molex connector at one end and a BNC (circular) socket at the other end which can be installed in the rear panel of a computer. When ordering, refer to it as the "VBC" cable. Vector Graphic computers have cutouts at the rear which are the right size for this socket.

3.4.3 Connecting a Keyboard

The keyboard is connected via a 24-pin dip plug to socket J1 on the right side of the board. The following diagram shows the pin assignments:

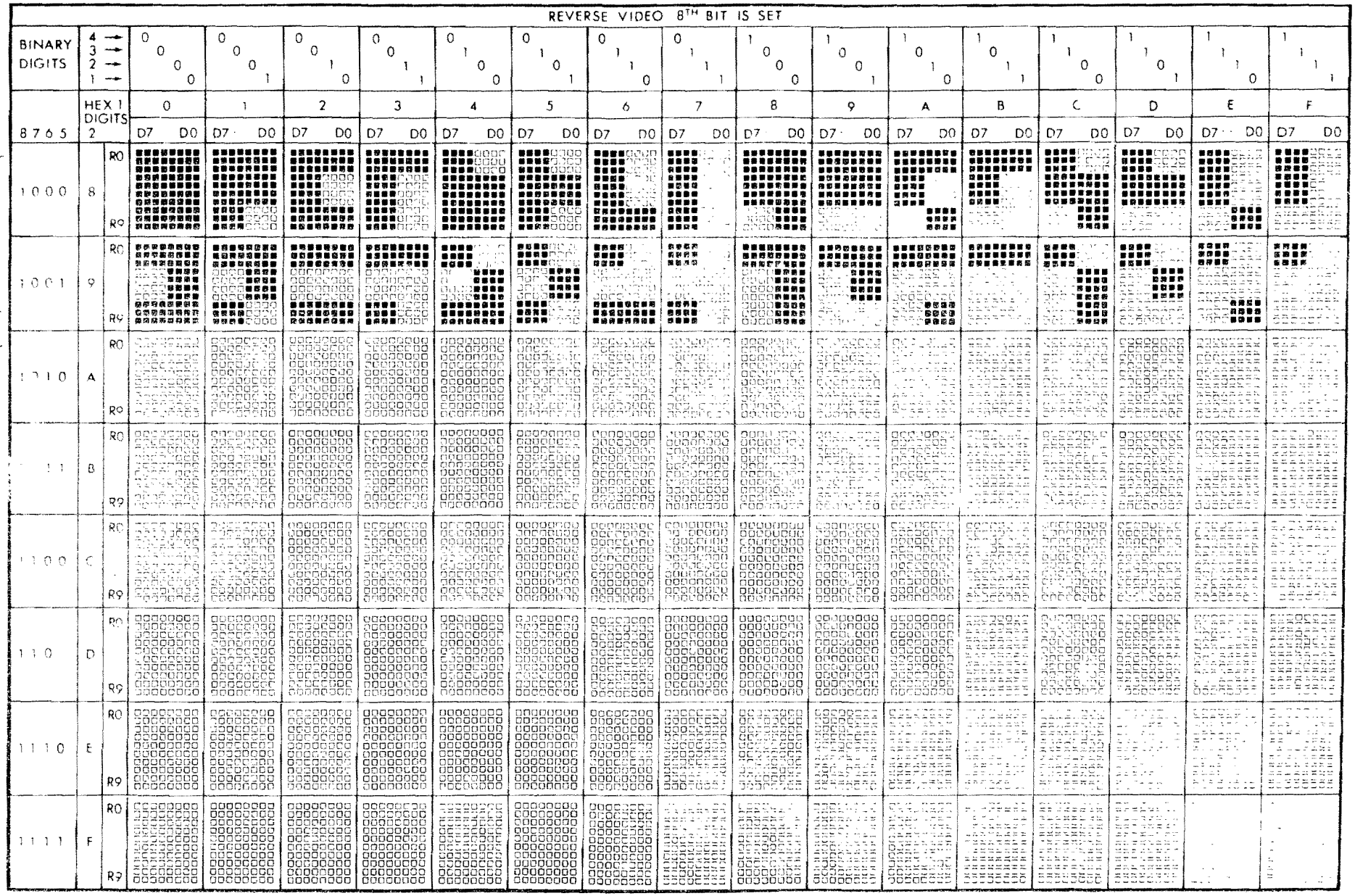


The cable shipped with the Vector Graphic Mindless Terminal has a properly wired 24-pin dip plug already attached. Simply plug it in. The use of pins 9 and 11 are discussed in sections 3.1.12 and 3.1.13.


NORMAL VIDEO
BINARY DIGITS
4 →
3 →
2 →
1 →
HEX 1 DIGITS
8 7 6 5
2
0 1 2 3 4 5 6 7 8 9 A B C D E F
D7 ··· D0
R0
R9
0000
0001
0010
0011
0100
0101
0110
0111

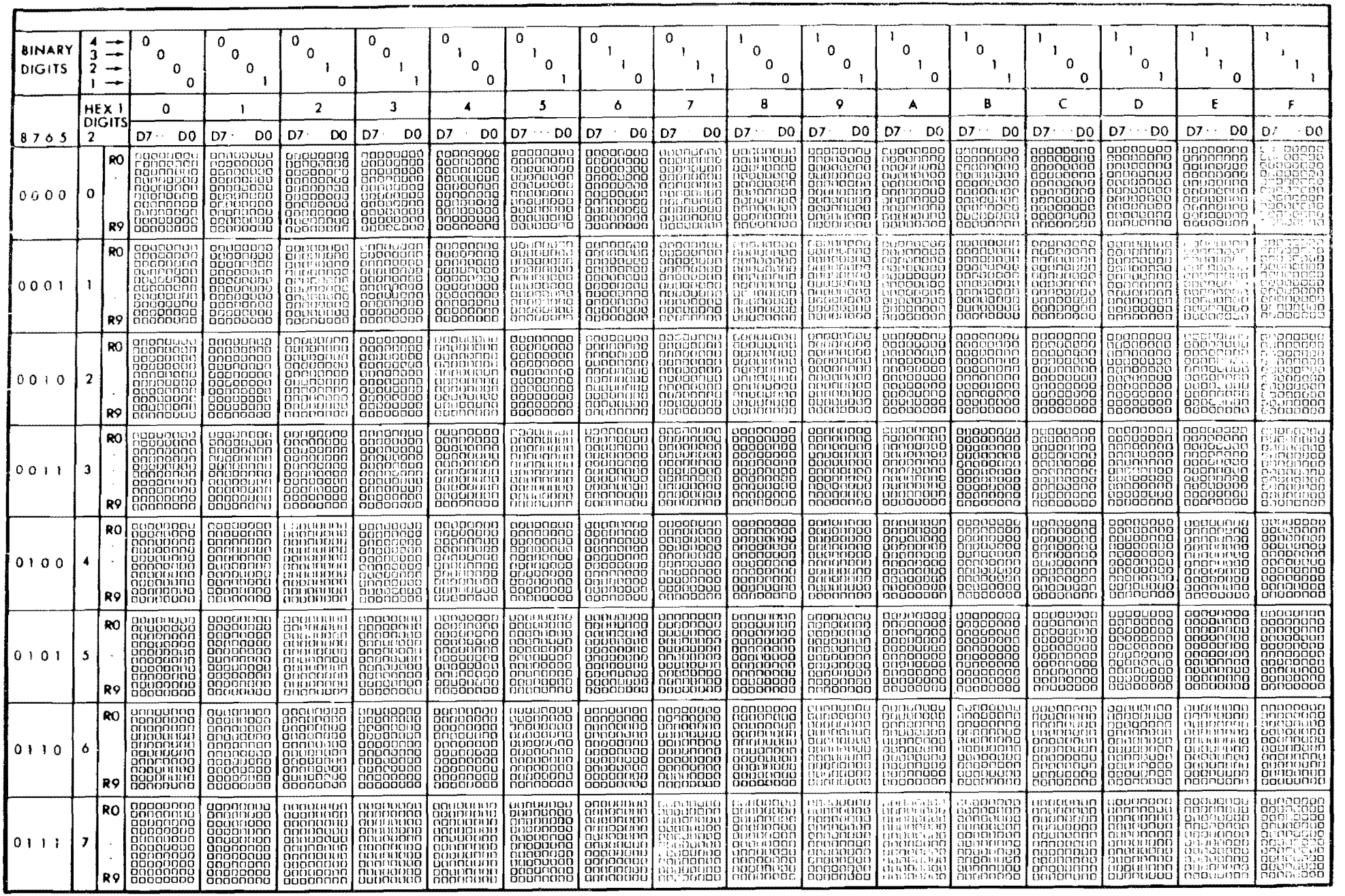
Vector Graphic Flashwriter II Board

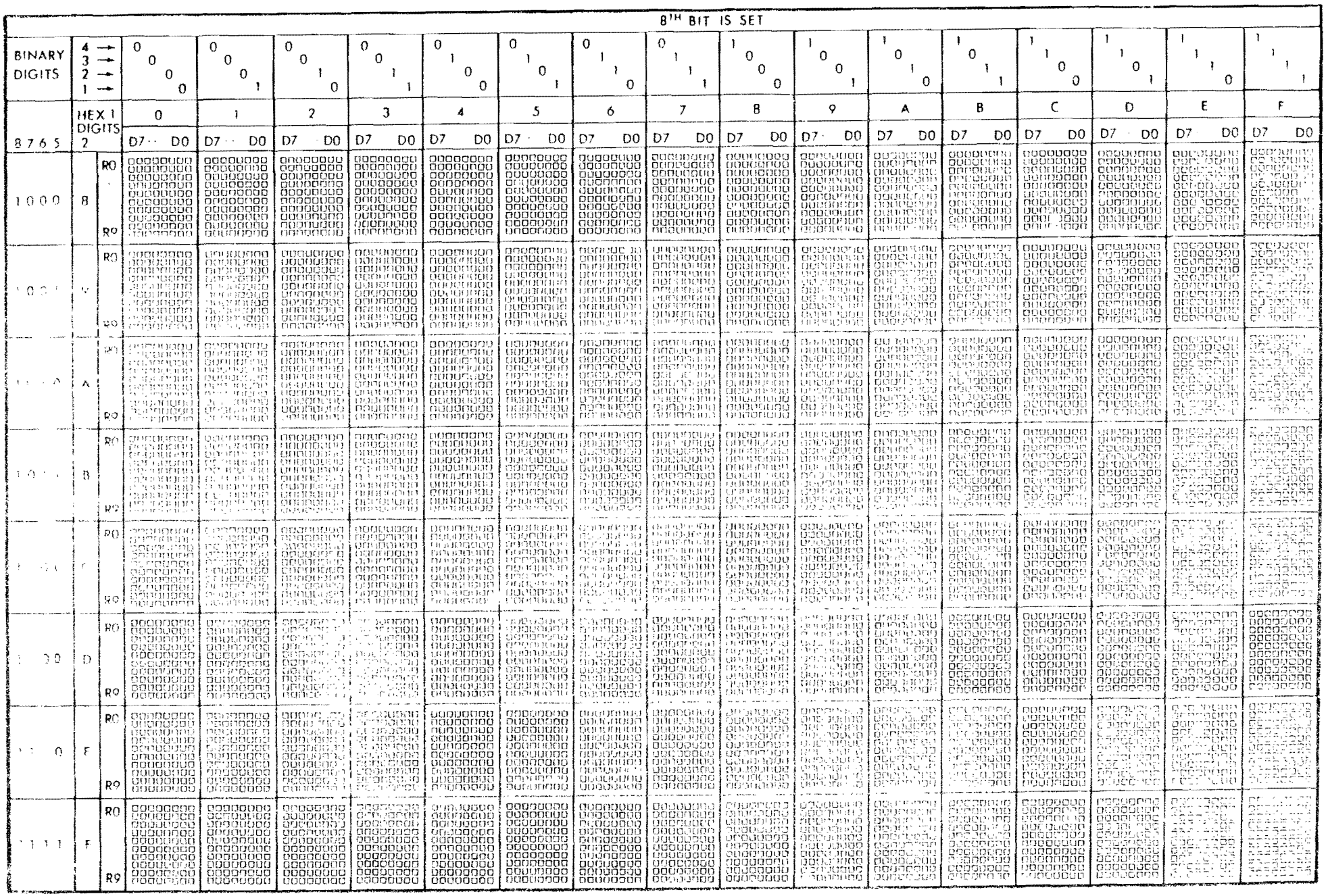
Revision 2 2/7/79



* REVERSE VIDEO VERSIONS OF GRAPHICS CHARACTERS

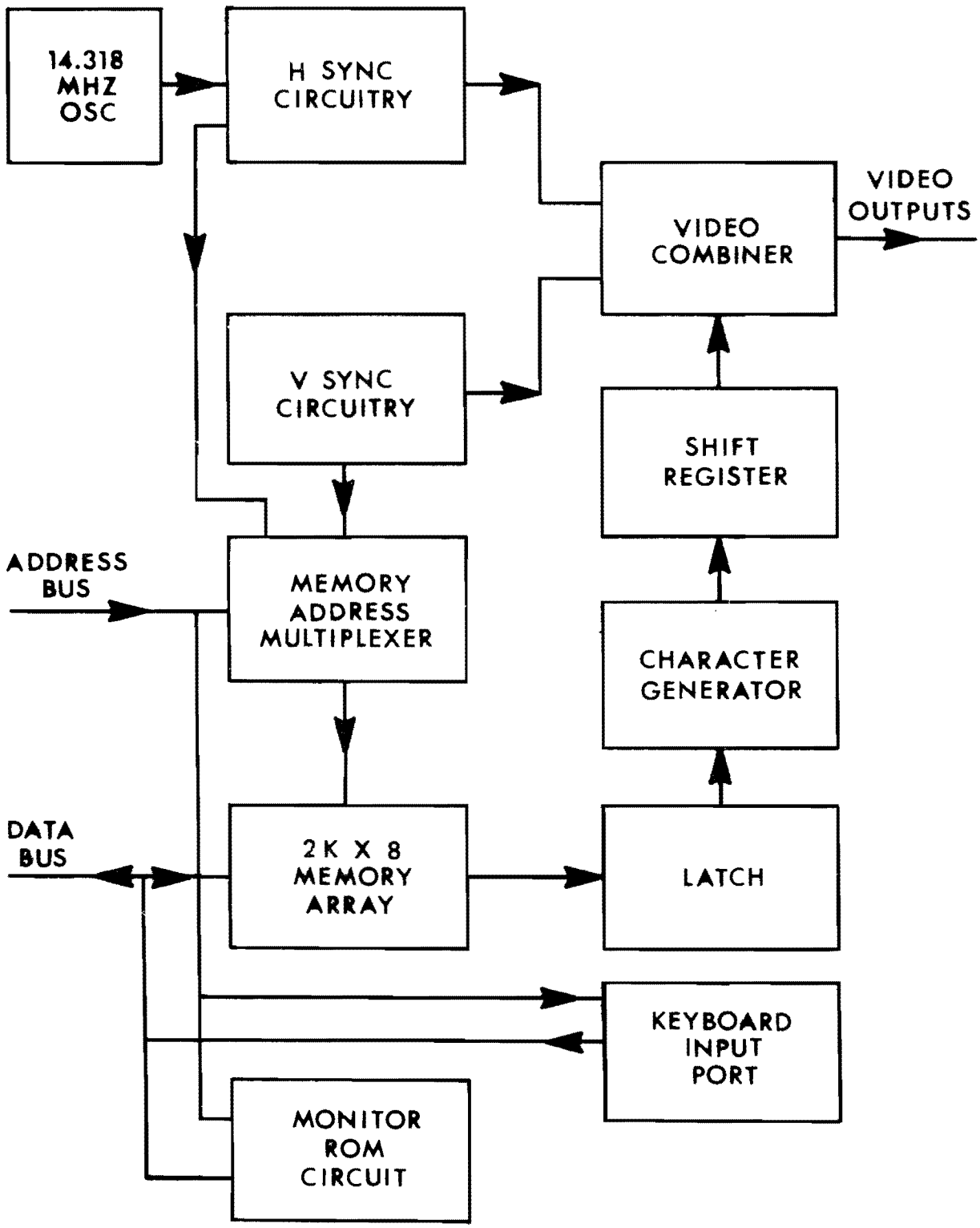
** REVERSE VIDEO VERSIONS OF ASCII CHARACTERS


BINARY DIGITS
4 →
3 →
2 →
1 →
HEX 1 DIGITS
8 7 6 5
2
D7 D0
0 0 0 0
0 0 0 1
0 0 1 0
0 0 1 1
0 1 0 0
0 1 0 1
0 1 1 0
0 1 1 1
1 0 0 0
1 0 0 1
1 0 1 0
1 0 1 1
1 1 0 0
1 1 0 1
1 1 1 0
1 1 1 1
0
1
2
3
4
5
6
7
8
9
A
B
C
D
E
F
0 0 0 0
0 0 0 1
0 0 1 0
0 0 1 1
R0
R9

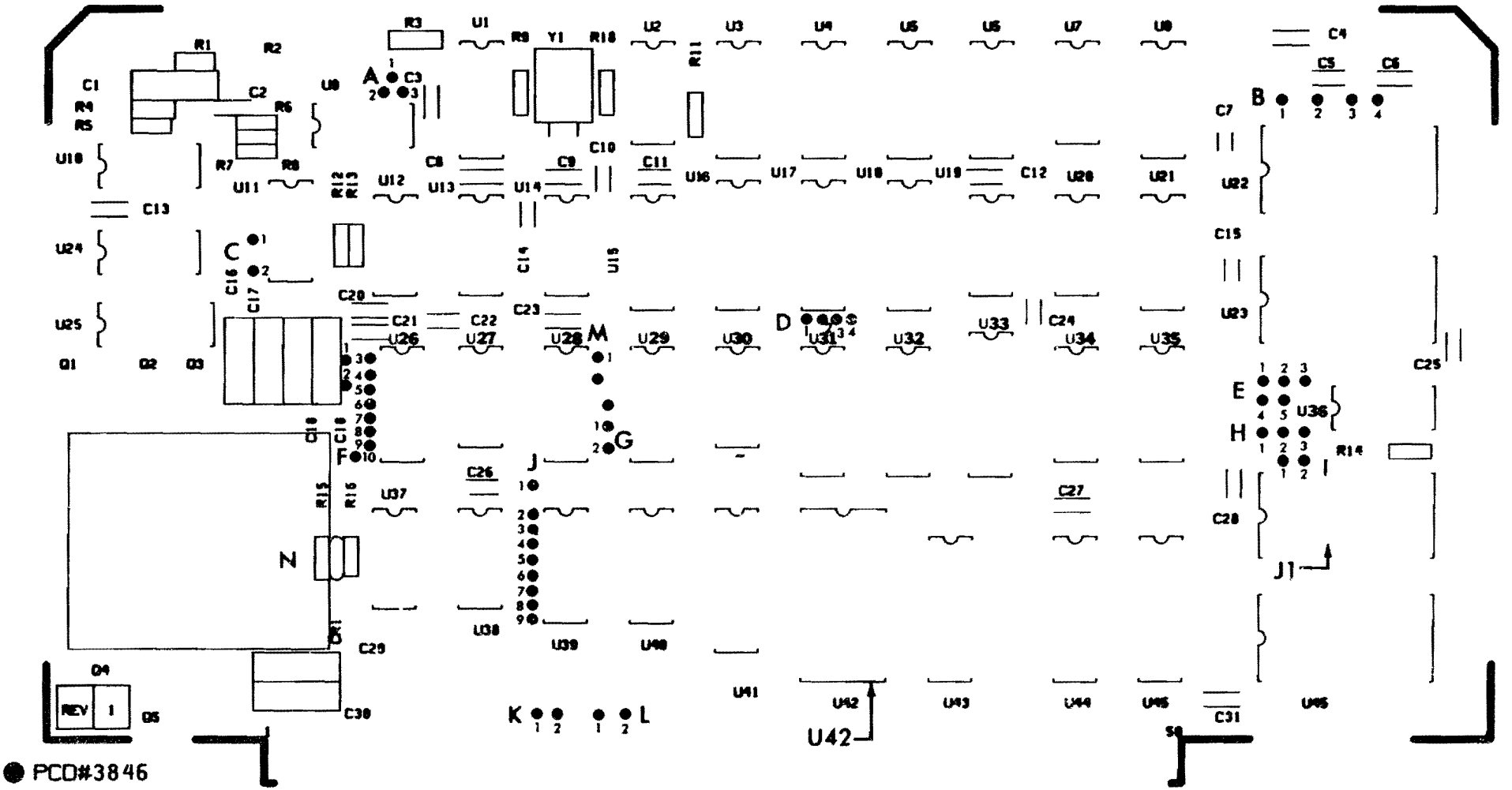


* REVERSE VIDEO VERSIONS OF GRAPHICS CHARACTERS

** REVERSE VIDEO VERSIONS OF ASCII CHARACTERS



80 X 24 VIDEO DISPLAY MODULE BLOCK DIAGRAM



SILK SCREEN
80X24 DISPLAY BOARD
VECTOR GRAPHIC

Revision 2 2/7/79

```
0000  E000 =  BASE        EQU   0E000H      ;ASSEMBLY ADDRESS
0000  E000 =  PR          EQU   0E000H      ;PROM/RAM ADDRESS
0000                      LINK  'M6'
0000          ************************************************
0000          *                                              *
0000          *        VECTOR MZ MONITOR - VERSION 4.2       *
0000          *     R. S. HAMP  7/16/79  MODIFIED 6/1/80     *
0000          *                                              *
0000          ************************************************
0000          *
0000          * SYSTEM EQUATES
0000  0000 =  CONS        EQU   0           ;CONS STATUS PRT
0000  0001 =  COND        EQU   1           ;CONS DATA PORT
0000  0040 =  RDA         EQU   40H         ;RECEIVE FLAG
0000  0000 =  STPOL       EQU   0           ;STATUS POLARITY
0000  FFD0 =  SPTR        EQU   PR+01FD0H   ;STACK POINTER
0000  E800 =  DSBOOT      EQU   0E800H      ;DUALSTOR BOOTSTRAP
0000  E802 =  MSBOOT      EQU   0E802H      ;MEGASTOR BOOTSTRAP
0000  E80C =  FLBOOT      EQU   0E80CH      ;FLOPPY BOOTSTRAP
0000  FF10 =  DBUSY       EQU   0FF10H      ;CONTROLLER BUSY
0000          *
0000          ********** COMMAND FORMAT **********
0000          *    A SSSS FFFF ASCII DUMP OF MEMORY          *
0000          *    B JUMP TO BOOTSTRAP LOADER                *
0000          *    C SSSS FFFF CCCC COMPARE BLOCKS           *
0000          *    D SSSS FFFF DUMP MEMORY IN HEX & ASCII    *
0000          *    E EXTERNAL COMMUNICATIONS                 *
0000          *    F SSSS FFFF DD DD TWO BYTE SEARCH         *
0000          *    G SSSS GO TO AND EXECUTE                  *
0000          *    H JUMP TO HIGH RAM AT FC00                *
0000          *    I PP INPUT FROM PORT                      *
0000          *    J JUMP TO DOS                             *
0000          *    K LLLL SET A BREAKPOINT                   *
0000          *    L JUMP TO LOW RAM AT 0                    *
0000          *    M SSSS FFFF DDDD MOVE BLOCK               *
0000          *    N NON DESTRUCTIVE MEMORY TEST             *
0000          *    O PP DD OUTPUT TO PORT                    *
0000          *    P LLLL PROGRAM MEMORY                     *
0000          *    Q SSSS FFFF COMPUTE CHECKSUM              *
0000          *    R DUMP Z-80 REGISTERS                     *
0000          *    S SSSS FFFF DD SEARCH FOR SINGLE BYTE     *
0000          *    T SSSS FFFF TEST MEMORY                   *
0000          *    U JUMP TO USER AREA AT 2B00               *
0000          *    V BOOT FROM 8 INCH DISK                   *
0000          *    W BOOT WINCHESTER DISK                    *
0000          *    X SSSS FFFF DDDD EXCHANGE BLOCK           *
0000          *    Y KEYBOARD ECHO                           *
0000          *    Z SSSS FFFF DD ZERO OR FILL MEMORY        *
0000          ************************************************
0000          *
0000                      ORG   BASE
E000          * JUMP TABLE OF ENTRY POINTS
E000 C315E0   MONIT       JMP   INIT        ;INITIALIZE ALL
E003 C31CE1   KEYST       JMP   KEYSTAT     ;TEST KEYBOARD
E006 C341E1   KEYDATA     JMP   CONVERT     ;INPUT KEYBOARD
E009 C378E3   CHP         JMP   VIDEO       ;OUTPUT TO SCREEN
E00C C32FE1   ESC         JMP   ESCAPE      ;KEYBOARD INPUT
```



```
E00F          *
E00F          * TABLE OF COMMANDS FOR USART
E00F 00000040 INITABLE    DB    0,0,0,40H,0CEH,27H
E013 CE27
E015          *
E015 31D0FF   INIT        LXI   SP,SPTR     ;INIT STACK
E018 CD2FE1               CALL  ESCAPE      ;DUMP LATCH
E01B AF                   XRA   A
E01C 32EAFF               STA   XYFLAG
E01F 3210FF               STA   DBUSY       ;CLEAR CONTROLLER FLAG
E022          * INITIALIZE USARTS AT PORTS 3,5,7
E022 0E03                 MVI   C,3         ;STARTING PORT
E024 0606     INILOOP     MVI   B,6         ;NO OF COMMANDS
E026 210FE0               LXI   H,INITABLE
E029 EDB3                 OUTIR             ;BLOCK OUTPUT
E02B 0C                   INR   C
E02C 0C                   INR   C
E02D 79                   MOV   A,C
E02E FE09                 CPI   9           ;DO 3 PORTS
E030 20F2                 JRNZ  INILOOP
E032          * PATCH RST 7
E032 3EC3                 MVI   A,0C3H      ;JUMP
E034 323800               STA   38H         ;RST 7
E037 21C8E6               LXI   H,DUMPREGS
E03A 223900               SHLD  39H
E03D          * DISPLAY SIGN ON
E03D CDCFE4               CALL  SIGN
E040          * CLEAR BREAKPOINT
E040 2AE7FF   CLRBRK      LHLD  BKPTLOC
E043 11E9FF               LXI   D,BRKCODE
E046 ED53E7FF             SDED  BKPTLOC
E04A 1A                   LDAX  D
E04B 77                   MOV   M,A
E04C 31D0FF   START       LXI   SP,SPTR     ;INITIALIZE STACK
E04F 2100F0               LXI   H,PAGE      ;FULL SCREEN SCROLL
E052 22DFFF               SHLD  TOSCN
E055 CD2BE5               CALL  PROMPT
E058 CD2FE1   KEYPOL      CALL  ESCAPE      ;READ KEYBOARD
E05B 38FB                 JRZ   KEYPOL
E05D E65F                 ANI   5FH         ;UPPER AND LOWER
E05F 214CE0               LXI   H,START
E062 E5                   PUSH  H
E063 FE04                 CPI   'D'-64
E065 CC7BE3               CZ    VIDEO       ;ECHO CLEARSCN
E068 FE41                 CPI   'A'
E06A D8                   RC                ;TOO SMALL
E06B FE5B                 CPI   05BH
E06D D0                   RNC               ;TOO LARGE
E06E 21F9E0               LXI   H,CMDTB+7EH
E071 F5                   PUSH  PSW
E072 87                   ADD   A
E073 85                   ADD   L
E074 6F                   MOV   L,A
E075 5E                   MOV   E,M
E076 23                   INX   H
E077 56                   MOV   D,M
E078 EB                   XCHG
```

```
E079 F1                     POP     PSW
E07A F9                     PCHL                    ;AWAY WE GO
E07B            * COMMAND TABLE
E07B 34E5       CMDTB       DW      WASCII          ;A
E07D 0CE8                   DW      FLYBOOT         ;B
E07F E2E2                   DW      COMPR           ;C
E081 B8E5                   DW      HEXRUL          ;D
E083 C0E7                   DW      EXTCOM          ;E
E085 05E3                   DW      FIND            ;F
E087 AFE0                   DW      EXEC            ;G
E089 56E2                   DW      RAM             ;H
E08B 53E3                   DW      PINPT           ;I
E08D 96E1                   DW      WARM            ;J
E08F 02E7                   DW      SETBRK          ;K
E091 62E2                   DW      LORAM           ;L
E093 96E2                   DW      MOVEB           ;M
E095 BEE2                   DW      NEXT            ;N
E097 65E3                   DW      POUTP           ;O
E099 05E6                   DW      PROGRAM         ;P
E09B 79E1                   DW      CHKSM           ;Q
E09D BCE6                   DW      DREGS           ;R
E09F 12E3                   DW      SRCH            ;S
E0A1 C3E1                   DW      TMEM            ;T
E0A3 47E2                   DW      USER            ;U
E0A5 00E8                   DW      DSBOOT          ;V
E0A7 02E8                   DW      MSBOOT          ;W
E0A9 87E2                   DW      EXCHG           ;X
E0AB AEE1                   DW      ECHO            ;Y
E0AD 6EE2                   DW      ZEROM           ;Z
E0AF            *
E0AF            *** EXECUTE THE PROGRAM AT THE ADDRESS ***
E0AF            *
E0AF CDC4E4     EXEC        CALL    PTSTNG
E0B2 474F2054               DTH     'GO TO '
E0B6 4FA0
E0B8 CDBDE0                 CALL    AHEX            ;READ ADD FROM KB
E0BB EB                     XCHG
E0BC E9                     PCHL                    ;JUMP TO IT
E0BD            *
E0BD            *** CONVERT UP TO 4 HEX DIGITS TO BIN
E0BD            *
E0BD 0E04       AHEX        MVI     C,4             ;COUNT OF 4 DIGITS
E0BF 210000     AHE0        LXI     H,0             ;16 BIT ZERO
E0C2 CD2FE1     AHE1        CALL    ESCAPE
E0C5 FE20                   CPI     ' '             ;SPACE?
E0C7 CAF8E0                 JZ      SPCOVR
E0CA CDBDE0                 CALL    HEX             ;CHECK VALUE
E0CD 38F3                   JRC     AHE1
E0CF 29                     DAD     H               ;MULT H*16
E0D0 29                     DAD     H
E0D1 29                     DAD     H
E0D2 29                     DAD     H
E0D3 85                     ADD     L
E0D4 6F                     MOV     L,A
E0D5 0D                     DCR     C               ;4 DIGITS?
E0D6 C2C2E0                 JNZ     AHE1            ;KEEP READING
E0D9 EB                     XCHG
```



```
E0DA 3E20       SPCE        MVI     A,20H           ;PRINT SPACE
E0DC C37BE3     PTCN        JMP     VIDEO
E0DF 3E0D       CRLF        MVI     A,0DH           ;PRINT CR
E0E1 CDDCE0                 CALL    PTCN
E0E4 3E0A                   MVI     A,0AH
E0E6 18F4                   JR      PTCN
E0E8            *
E0E8 CD7BE3     SPCOVR      CALL    VIDEO
E0EB 18EC                   JR      SPCE-1
E0ED            *
E0ED            * CHECK FOR HEX VALUE, CONVERT
E0ED FE30       HEX         CPI     30H             ;<0
E0EF D8                     RC
E0F0 FE3A                   CPI     ':'             ;>9
E0F2 3809                   JRC     NUM
E0F4 E65F                   ANI     5FH             ;UPPER & LOWER CASE
E0F6 FE41                   CPI     'A'             ;<A
E0F8 D8                     RC
E0F9 FE47                   CPI     'G'             ;>F
E0FB 3F                     CMC
E0FC D8                     RC
E0FD CD7BE3     NUM         CALL    VIDEO
E100 D630                   SUI     48              ;ASCII BIAS
E102 FE0A                   CPI     10              ;DIGIT 0-10
E104 3802                   JRC     ALFA
E106 D607                   SUI     7               ;ALPHA BIAS
E108 A7         ALFA        ANA     A               ;CLEAR CY
E109 C9                     RET                     ;WITH CY CLEAR
E10A            *
E10A            * READ 2 DIGITS FROM THE CONSOLE
E10A 0E02       AHE2        MVI     C,2
E10C 18B1                   JR      AHE0
E10E            *
E10E            * SHORT ROUTINE TO SAVE CODE
E10E CDBDE0     TAHEX       CALL    AHEX
E111 18AA                   JR      AHEX
E113            *
E113            *** READ FROM CONSOLE TO REG A ***
E113            *
E113 CD2FE1     RDCN        CALL    ESCAPE          ;READ KEYBOARD
E116 28FB                   JRZ     RDCN
E118 FE60                   CPI     60H
E11A 38C0                   JRC     PTCN
E11C E65F                   ANI     5FH
E11E 18BC                   JR      PTCN
E120            *
E120 CD2FE1     PAUSE       CALL    ESCAPE
E123 FE20                   CPI     20H
E125 C0                     RNZ
E126 CD2FE1     PLOOP       CALL    ESCAPE
E129 FE20                   CPI     20H
E12B C226E1                 JNZ     PLOOP
E12E C9                     RET
E12F            *
E12F CD3CE1     ESCAPE      CALL    KEYSTAT
E132 C8                     RZ
E133 CD41E1                 CALL    CONVERT
```

```
E136 FE1B               CPI     1BH             ;ESCAPE
E138 CA4CE0             JZ      START
E13B C9                 RET
E13C            *
E13C DB00       KEYSTAT IN      CONS
E13E E640               ANI     80A
E140 C9                 RET
E141            *
E141            * KEYBOARD CODE CONVERSION
E141 DB01       CONVERT IN      COND            ;KEYBOARD DATA
E143 E5                 PUSH    H
E144 C5                 PUSH    B
E145 010500             LXI     B,TABLEND-KTABL/2
E148 2150E1             LXI     H,KTABL
E14B BDA1       LOOP    CCI                     ;COMPARE TABLE
E14D 2806               JRZ     FND
E14F 23                 INX     H
E150 EA4BE1             JPE     LOOP            ;CONT LOOKING
E153 1801               JR      NFND
E155 7E         FND     MOV     A,M             ;NEW CODE
E156 E67F       NFND    ANI     7FH             ;MASK DOWN
E158 C1                 POP     B
E159 E1                 POP     H
E15A C9                 RET
E15B            *
E15B            * THIS TABLE CAN BE EXTENDED IF DESIRED
E15B E15D       KTABL   DD      0E15DH          ;|
E15D F15B               DD      0F15BH          ;[
E15F A17E               DD      0A17EH          ;~
E161 B15C               DD      0B15CH          ;•
E163 6015               DD      06015H          ;CURSOR UP
E165   E165 =   TABLEND EQU     $
E165                    ORG     KTABL+30        ;ROOM FOR 15 CONVS
E179            *
E179            * CHECKSUM ROUTINE
E179 CDC4E4     CHKSM   CALL    PTSTNG
E17C 43484543           DTH     'CHECKSUM '
E180 4053554D
E184 A0
E185 CD0FE1             CALL    TAHEX
E188 0600               MVI     B,0
E18A 7E         CHKSMLP MOV     A,M
E18B 80                 ADD     B
E18C 47                 MOV     B,A
E18D CD3FE2             CALL    BMP
E190 20F8               JRNZ    CHKSMLP
E192 78                 MOV     A,B
E193 C326E2             JMP     PT2
E196            *
E196            * WARM START
E196            *
E196 CDC4E4     WARM    CALL    PTSTNG
E199 4A554D50           DTH     'JUMP TO DOS'
E19D 20544F20
E1A1 444F53
E1A4 21E704             LXI     H,04E7H         ;MDOS RESTART
E1A7 7E                 MOV     A,M
```



```
E1A8 FEC3               CPI     0C3H
E1AA C20000             JNZ     0               ;CP/M RESTART
E1AD E9                 PCHL                    ;MDOS WARM START
E1AE            *
E1AE            * KEYBOARD ECHO ROUTINE
E1AE CDC4E4     ECHO    CALL    PTSTNG
E1B1 4543484F           DTH     'ECHO KEYS '
E1B5 20404559
E1B9 53A0
E1BB CD2FE1     ECOLP   CALL    ESCAPE          ;LOOK AT KEYBOARD
E1BE C4DCE0             CNZ     PTCN            ;PRINT IF KEYPRESS
E1C1 18F8               JR      ECOLP           ;CONTINUE LOOPING
E1C3            *
E1C3            *** MEMORY TEST ROUTINE ***
E1C3            *
E1C3 CDC4E4     TMEM    CALL    PTSTNG
E1C6 54455354           DTH     'TEST '
E1CA A0
E1CB CD0EE1             CALL    TAHEX           ;READ ADDRESSES
E1CE 015A5A             LXI     B,5A5AH         ;INI B,C
E1D1 CDFDE1     CYCL    CALL    RNDM
E1D4 C5                 PUSH    B               ;KEEP ALL REGS
E1D5 E5                 PUSH    H
E1D6 D5                 PUSH    D
E1D7 CDFDE1     TLOP    CALL    RNDM
E1DA 70                 MOV     M,B             ;WRITE IN MEM
E1DB CD3FE2             CALL    BMP
E1DE C2D7E1             JNZ     TLOP            ;REPEAT LOOP
E1E1 D1                 POP     D
E1E2 E1                 POP     H               ;RESTORE ORIG
E1E3 C1                 POP     B               ;VALUES OF
E1E4 E5                 PUSH    H
E1E5 D5                 PUSH    D
E1E6 CDFDE1     RLOP    CALL    RNDM            ;GEN NEW SEQ
E1E9 7E                 MOV     A,M             ;READ MEM
E1EA B8                 CMP     B               ;COMP MEM
E1EB C41DE2             CNZ     ERR             ;CALL ERROR RTN
E1EE CD3FE2             CALL    BMP
E1F1 C2E6E1             JNZ     RLOP
E1F4 D1                 POP     D
E1F5 E1                 POP     H
E1F6 3E2E               MVI     A,'.'
E1F8 CD7BE3             CALL    VIDEO
E1FB 18D4               JR      CYCL
E1FD            *** THIS ROUTINE GENERATES RANDOM NOS ***
E1FD CD20E1     RNDM    CALL    PAUSE
E200 78                 MOV     A,B             ;LOOK AT B
E201 E6B4               ANI     0B4H            ;MASK BITS
E203 A7                 ANA     A               ;CLEAR CY
E204 EA08E2             JPE     PEVE            ;JUMP IF EVEN
E207 37                 STC
E208 79         PEVE    MOV     A,C             ;LOOK AT C
E209 17                 RAL                     ;ROTATE CY IN
E20A 4F                 MOV     C,A             ;RESTORE C
E20B 78                 MOV     A,B             ;LOOK AT B
E20C 17                 RAL                     ;ROTATE CY IN
E20D 47                 MOV     B,A             ;RESTORE B
```

```
E20E C9                         RET                     ;RETURN W NEW B,C
E20F                    *
E20F                    *** ERROR PRINT OUT ROUTINE
E20F                    *
E20F CD0FE0     PTAD            CALL    CRLF            ;PRINT CR,LF
E212 CD20E1                     CALL    PAUSE
E215 7C                         MOV     A,H             ;PRINT
E216 CD26E2                     CALL    PT2             ;ASCII
E219 7D                         MOV     A,L             ;CODES
E21A C31CE7                     JMP     PT2S            ;FOR ADDRESS
E21D                    *
E21D F5         ERR             PUSH    PSW             ;SAVE ACC
E21E CD0FE2                     CALL    PTAD            ;PRINT ADD.
E221 78                         MOV     A,B             ;DATA
E222 CD1CE7                     CALL    PT2S            ;WRITTEN
E225 F1                         POP     PSW             ;DATA READ
E226 F5         PT2             PUSH    PSW
E227 CD2DE2                     CALL    BINH
E22A F1                         POP     PSW
E22B 1804                       JR      BINL
E22D 1F         BINH            RAR                     ;SHIFT RHT 4 BITS
E22E 1F                         RAR
E22F 1F                         RAR
E230 1F                         RAR
E231 E60F       BINL            ANI     0FH             ;LOW 4 BITS
E233 C630                       ADI     48              ;ASCII BIAS
E235 FE3A                       CPI     58              ;DIGIT 0-9
E237 DADCE0                     JC      PTCN
E23A C607                       ADI     7               ;DIGIT A-F
E23C C3DCE0                     JMP     PTCN
E23F                    *
E23F                    * COMPARE ADDRESSES AND INCREMENT H
E23F 7B         BMP             MOV     A,E
E240 95                         SUB     L
E241 2002                       JRNZ    GOON
E243 7A                         MOV     A,D
E244 9C                         SBB     H
E245 23         GOON            INX     H
E246 C9                         RET
E247                    *
E247                    * JUMP TO USER RAM
E247 CDC4E4     USER            CALL    PTSTNG
E24A 55534552                   DTH     'USER AREA'
E24E 20415245
E252 C1
E253 C30001                     JMP     01001
E256                    *
E256                    * JUMP TO RAM AT PR+1C00
E256 CDC4E4     RAM             CALL    PTSTNG
E259 48492052                   DTH     'HI RAM'
E25D 41CD
E25F C300FC                     JMP     PR+1C001
E262                    *
E262                    * JUMP TO RAM AT 0
E262 CDC4E4     LORAM           CALL    PTSTNG
E265 4C4F2052                   DTH     'LO RAM'
E269 41CD
```



```
E26B C30000                     JMP     0
E26E                    *
E26E                    * ZERO OR FILL MEMORY WITH A CONSTANT
E26E CDC4E4     ZEROM           CALL    PTSTNG
E271 46494C4C                   DTH     'FILL '
E275 A0
E276 CD0EE1                     CALL    TWHEX           ;READ ADDRESSES
E279 E5                         PUSH    H               ;SAVE H
E27A CD0AE1                     CALL    AHE2            ;READ 2 DIGITS
E27D EB                         XCHG
E27E E3                         XTHL                    ;RESTORE H,L
E27F C1                         POP     B
E280 71         ZLOOP           MOV     M,C             ;WRITE INTO MEM
E281 CD3FE2                     CALL    BMP             ;COMP ADD, INCR H
E284 C8                         RZ                      ;RETURN IF DONE
E285 18F9                       JR      ZLOOP           ;CONTINUE TIL DONE
E287                    * EXCHANGE OR MOVE A BLOCK OF MEMORY
E287 47         EXCHG           MOV     B,A
E288 CDC4E4                     CALL    PTSTNG
E28B 45584348                   DTH     'EXCHANGE '
E28F 414E4745
E293 A0
E294 1809                       JR      MOVENTR
E296 47         MOVEB           MOV     B,A             ;SAVE CODE
E297 CDC4E4                     CALL    PTSTNG
E29A 4D4F5645                   DTH     'MOVE '
E29E A0
E29F CD0EE1     MOVENTR         CALL    TWHEX           ;READ ADDRESSES
E2A2 E5                         PUSH    H
E2A3 CD0DE0                     CALL    AHEX
E2A6 EB                         XCHG
E2A7 E3                         XTHL                    ;BACK TO NORMAL
E2A8 4E         MLOOP           MOV     C,M
E2A9 E3                         XTHL
E2AA 78                         MOV     A,B
E2AB FE4D                       CPI     'M'
E2AD 2804                       JRZ     NEXCH
E2AF 7E                         MOV     A,M
E2B0 E3                         XTHL
E2B1 77                         MOV     M,A
E2B2 E3                         XTHL
E2B3 71         NEXCH           MOV     M,C
E2B4 23                         INX     H
E2B5 E3                         XTHL
E2B6 CD3FE2                     CALL    BMP
E2B9 CA4CE0                     JZ      START
E2BC 18EA                       JR      MLOOP
E2BE                    * NON DESTRUCTIVE MEMORY TEST
E2BE CDC4E4     NDMT            CALL    PTSTNG
E2C1 4D454D20                   DTH     'MEM CHECK'
E2C5 43484543
E2C9 CB
E2CA 210000                     LXI     H,0             ;START AT ZERO
E2CD 4E         NDLOP           MOV     C,M
E2CE 06FF                       MVI     B,0FFH
E2D0 70                         MOV     M,B
E2D1 7E                         MOV     A,M
```

```
E2D2 B8                         CMP     B
E2D3 C2DDE2                     JNZ     ERRJP           ;PRINT ERROR
E2D6 0600                       MVI     B,0
E2D8 70                         MOV     M,B
E2D9 7E                         MOV     A,M
E2DA B8                         CMP     B
E2DB C21DE2     ERRJP           JNZ     ERR
E2DE 71                         MOV     M,C
E2DF 23                         INX     H
E2E0 18EB                       JR      NDLOP
E2E2            * COMPARE TWO BLOCKS OF MEMORY
E2E2 CDC4E4     COMPR           CALL    PTSTNG
E2E5 434F4D50                   DTH     'COMPARE '
E2E9 415245A0
E2ED CD0EE1                     CALL    TAHEX
E2F0 E5                         PUSH    H
E2F1 CD0DE0                     CALL    AHEX
E2F4 EB                         XCHG
E2F5 7E         VMLOP           MOV     A,M
E2F6 23                         INX     H
E2F7 E3                         XTHL
E2F8 BE                         CMP     M
E2F9 46                         MOV     B,M
E2FA C41DE2                     CNZ     ERR
E2FD CD3FE2                     CALL    BMP
E300 E3                         XTHL
E301 20F2                       JRNZ    VMLOP
E303 F1                         POP     PSW
E304 C9                         RET
E305            * SEARCH FOR SPECIFIC CODES
E305 F5         FIND            PUSH    PSW
E306 CDC4E4                     CALL    PTSTNG
E309 46494E44                   DTH     'FIND-2 '
E30D 2D32A0
E310 180D                       JR      SRCHENT
E312 F5         SRCH            PUSH    PSW
E313 CDC4E4                     CALL    PTSTNG
E316 53454152                   DTH     'SEARCH-1 '
E31A 43482D31
E31E A0
E31F CD0EE1     SRCHENT         CALL    TAHEX
E322 E5                         PUSH    H               ;SAVE H
E323 CD0AE1                     CALL    AHE2            ;READ 2 DIGITS
E326 EB                         XCHG                    ;H=CODE,D=P
E327 45                         MOV     B,L             ;PUT CODE IN B
E328 E1                         POP     H               ;RESTORE H
E329 F1                         POP     PSW
E32A FE53                       CPI     'S'
E32C F5                         PUSH    PSW
E32D 2807                       JRZ     CONT
E32F E5                         PUSH    H
E330 CD0AE1                     CALL    AHE2            ;READ 2 DIGITS
E333 EB                         XCHG
E334 4D                         MOV     C,L
E335 E1                         POP     H
E336 7E         CONT            MOV     A,M             ;READ MEMORY
E337 B8                         CMP     B               ;COMPARE TO CODE
```



```
E338 2012                       JRNZ    SKP             ;SKIP IF NO COMP
E33A F1                         POP     PSW             ;FETCH CONTROL
E33B FE53                       CPI     'S'
E33D F5                         PUSH    PSW
E33E 2806                       JRZ     OKCP
E340 23                         INX     H
E341 7E                         MOV     A,M
E342 2B                         DCX     H
E343 B9                         CMP     C
E344 2006                       JRNZ    SKP
E346 23         OKCP            INX     H
E347 7E                         MOV     A,M             ;READ NEXT BYTE
E348 2B                         DCX     H               ;DECR ADDRESS
E349 CD1DE2                     CALL    ERR             ;PRINT CODES
E34C CD3FE2     SKP             CALL    BMP             ;CHECK IF DONE
E34F 20E5                       JRNZ    CONT            ;BACK FOR MORE
E351 F1                         POP     PSW
E352 C9                         RET
E353            *
E353            * INPUT DATA FROM A PORT
E353 CDC4E4     PINPT           CALL    PTSTNG
E356 494E5055                   DTH     'INPUT '
E35A 54A0
E35C CD0AE1                     CALL    AHE2            ;READ 2 DIGITS
E35F 4B                         MOV     C,E
E360 ED78                       INP     A
E362 C326E2                     JMP     PT2
E365            *
E365            * OUTPUT TO A PORT
E365 CDC4E4     POUTP           CALL    PTSTNG
E368 4F555450                   DTH     'OUTPUT '
E36C 5554A0
E36F CD0AE1                     CALL    AHE2            ;READ 2 DIGITS
E372 CD0AE1                     CALL    AHE2            ;READ 2 DIGITS
E375 4D                         MOV     C,L
E376 ED59                       OUTP    E
E378 C9                         RET
E379            *
```

```
E379                    *
E379                    ****************************************
E379                    *                                      *
E379                    *     VIDEO DRIVER FOR  FLASHWRITER II *
E379                    *                                      *
E379                    ****************************************
E379                    *
E379    F000 =  PAGE            EQU     PR+1000H        ;SCREEN LOCATION
E379    0020 =  SPACE           EQU     20H
E379    0004 =  CLRSCRN         EQU     4
E379                    ****************************************
E379                    *                                      *
E379                    * CONTROL CODE COMMANDS:               *
E379                    *   (B) HOME CURSOR                    *
E379                    *   (D) CLEAR SCREEN                   *
E379                    *   (E) PRINT CONTROL CODE             *
E379                    *   (H) BACKSPACE                      *
E379                    *   (I) TAB                            *
E379                    *   (J) LINEFEED                       *
E379                    *   (M) CARRIAGE RETURN                *
E379                    *   (N) NO CURSOR                      *
E379                    *   (P) CLEAR TO END OF SCREEN         *
E379                    *   (Q) CLEAR TO END OF LINE           *
E379                    *   (R) CURSOR DOWN                    *
E379                    *   (T) TOGGLE REVERSE VIDEO           *
E379                    *   (U) CURSOR UP                      *
E379                    *   (W) CURSOR LEFT                    *
E379                    *   (X) CLEAR TO START OF LINE         *
E379                    *   (Z) CURSOR RIGHT                   *
E379                    *   ESC XY POSITION LEAD-IN            *
E379                    *                                      *
E379                    ****************************************
E379                    *
E379                    * VIDEO BOARD PARAMETERS
E379    0050 =  HORIZ           EQU     80              ;NO. OF CHARACTERS
E379    0018 =  VERT            EQU     24              ;NO. OF LINES
E379                    *
E379 3E14       TVIDEO          MVI     A,'T'-64        ;TOGGLE VIDEO
E37B                    *
E37B F5         VIDEO           PUSH    PSW
E37C C5                         PUSH    B
E37D D5                         PUSH    D
E37E E5                         PUSH    H
E37F E67F                       ANI     07FH
E381 4F                         MOV     C,A
E382 3A00E8                     LDA     BASE+8001H
E385 FEC3                       CPI     0C3H            ;PROM THERE?
E387 79                         MOV     A,C
E388 CC00E8                     CZ      BASE+8001H      ;CALL IT IF SO
E38B CD60E4     DISPL           CALL    LIFTCURS        ;ERASE CURSOR
E38E 3AEAFF                     LDA     XYFLAG
E391 A7                         ANA     A
E392 280A                       JRZ     NOXY
E394 3D                         DCR     A
E395 32EAFF                     STA     XYFLAG
E398 CAAFE4                     JZ      YPOS
E39B C3A6E4                     JMP     XPOS
```



```
E39E 79         NOXY            MOV     A,C             ;RECOVER CHARACTER
E39F FE20                       CPI     SPACE           ;PRINTING CODE?
E3A1 F2D5E3                     JP      PRINT
E3A4 FE1C                       CPI     PCL-TABL        ;TOO LARGE?
E3A6 F242E4                     JP      RET
E3A9 E5                         PUSH    H               ;CURSOR IN MEMORY
E3AA 21B8E3                     LXI     H,TABL          ;TABLE START
E3AD 5F                         MOV     E,A
E3AE 1600                       MVI     D,0
E3B0 19                         DAD     D
E3B1 5E                         MOV     E,M
E3B2 21D4E3                     LXI     H,PCL
E3B5 19                         DAD     D
E3B6 E3                         XTHL                    ;RECOVER H
E3B7 C9                         RET                     ;EXECUTE ROUTINE
E3B8                    * CONTROL CHARACTER JUMP TABLE
E3B8 6E         TABL            DB      RET-PCL         ;@
E3B9 6E                         DB      RET-PCL         ;A
E3BA 63                         DB      HOME-PCL        ;B HOME CURSOR
E3BB 6E                         DB      RET-PCL         ;C
E3BC 60                         DB      FORM-PCL        ;D CLEAR SCREEN
E3BD 00                         DB      PCL-PCL         ;E PRT CONTROL
E3BE 6E                         DB      RET-PCL         ;F
E3BF 6E                         DB      RET-PCL         ;G
E3C0 42                         DB      DBACKSP-PCL     ;H BACKSPACE
E3C1 59                         DB      TAB-PCL         ;I TAB OVER
E3C2 12                         DB      LINF-PCL        ;J LINE FEED
E3C3 6E                         DB      RET-PCL         ;K
E3C4 6E                         DB      RET-PCL         ;L
E3C5 6A                         DB      CRET-PCL        ;M CARRIAGE RET
E3C6 71                         DB      RET+3-PCL       ;N NO CURSOR
E3C7 6E                         DB      RET-PCL         ;O
E3C8 A7                         DB      CLEND-PCL       ;P CLR SCN TO END
E3C9 AC                         DB      CLLINE-PCL      ;Q CLR LINE TO END
E3CA 12                         DB      LINF-PCL        ;R CURSOR DOWN
E3CB 6E                         DB      RET-PCL         ;S
E3CC 76                         DB      TVIDF-PCL       ;T TOGGLE VIDEO
E3CD 80                         DB      CURSUP-PCL      ;U CURSOR UP
E3CE 6E                         DB      RET-PCL         ;V
E3CF 50                         DB      BACKSP-PCL      ;W CURSOR LEFT
E3D0 E4                         DB      CLSTRT-PCL      ;X CLR START OF LN
E3D1 6E                         DB      RET-PCL         ;Y
E3D2 06                         DB      EOL-PCL         ;Z CURSOR RIGHT
E3D3 CB                         DB      LEDIN-PCL       ;[ ESC=XY LEAD IN
E3D4                    *
E3D4                    * PRINT CODE IN B REGARDLESS
E3D4 48         PCL             MOV     C,B
E3D5                    * PRINT THE CHARACTER ON THE SCREEN
E3D5 3ADDFF     PRINT           LDA     VFL
E3D8 A9                         XRA     C
E3D9 77                         MOV     M,A
E3DA                    * EOL CHECKS THE CURS POS FOR END OF LINE
E3DA 3ADDFF     EOL             LDA     CURPOS
E3DD 3C                         INR     A
E3DE FE50                       CPI     HORIZ
E3E0 3005                       JRC     TABRET
E3E2 AF                         XRA     A
```

```
E3E3 32DBFF                     STA     CURPOS
E3E6                    * MOVE DN 1 LINE
E3E6 3ADCFF             LINF    LDA     LINENO
E3E9 FE17                       CPI     VERT-1
E3EB 2023                       JRNZ    NOSCRL
E3ED                    * SCROLL UP ONE LINE
E3ED 215000             SCROLL  LXI     H,HORIZ
E3F0 ED5BDFFF                   LDED    TOSCN
E3F4 19                         DAD     D
E3F5 EDA0               SCRL    LDI
E3F7 EDA0                       LDI
E3F9 7C                         MOV     A,H
E3FA FEF7                       CPI     HORIZ*VERT+PAGE/256
E3FC 20F7                       JRNZ    SCRL
E3FE 7D                         MOV     A,L
E3FF FE80                       CPI     HORIZ*VERT+PAGE&0FFH
E401 20F2                       JRNZ    SCRL
E403 3ADCFF                     LDA     LINENO
E406                    * ERASE BOTTOM LINE
E406 EB                 EBOTL   XCHG
E407 0650                       MVI     B,HORIZ
E409 3620               ELOP    MVI     M,SPACE
E40B 23                         INX     H
E40C 05                         DCR     B
E40D 20FA                       JRNZ    ELOP
E40F 3D                         DCR     A
E410 3C                 NOSCRL  INR     A
E411 32DCFF                     STA     LINENO
E414 182C                       JR      RET
E416                    *
E416                    * ERASE BEFORE BACKSPACING
E416 3620               DBACKSP MVI     M,20H
E418 3ADBFF                     LDA     CURPOS
E41B A7                         ANA     A
E41C 2824                       JRZ     RET
E41E 3D                         DCR     A
E41F 2B                         DCX     H
E420 3620                       MVI     M,20H
E422 181B                       JR      TABRET
E424                    * MOVE THE CURSOR BACK
E424 3ADBFF             BACKSP  LDA     CURPOS
E427 3D                         DCR     A
E428 F23FE4                     JP      TABRET
E42B 1811                       JR      CRET
E42D                    * TAB OVER TO THE NEXT 8 MULTIPLE
E42D 3ADBFF             TAB     LDA     CURPOS
E430 F607                       ORI     7
E432 18A9                       JR      EOL+3
E434                    * CLEAR THE SCREEN AND HOME UP
E434 CD8DE4             FORM    CALL    CLEAR
E437 AF                 HOME    XRA     A
E438 32DCFF                     STA     LINENO
E43B 32DDFF                     STA     VFL             ;CLR VID FLAG
E43E                    * CARRIAGE RETURN
E43E AF                 CRET    XRA     A
E43F 32DBFF             TABRET  STA     CURPOS
E442                    * RETURN TO THE CALLING ROUTINE
```



```
E442 CD60E4             RET     CALL    LIFTCURS
E445 E1                         POP     H
E446 D1                         POP     D
E447 C1                         POP     B
E448 F1                         POP     PSW
E449 C9                         RET
E44A 3ADDFF             TVIDF   LDA     VFL
E44D EE80                       XRI     80H
E44F 32DDFF                     STA     VFL
E452 18EE                       JR      RET
E454                    *
E454                    * MOVE THE CURSOR UP
E454 3ADCFF             CURSUP  LDA     LINENO
E457 A7                         ANA     A
E458 28E8                       JRZ     RET
E45A 3D                         DCR     A
E45B 32DCFF             STORLN  STA     LINENO
E45E 18E2                       JR      RET
E460                    * CALCULATE MEM ADD FROM CURSOR POSITION
E460 2180F7             LIFTCURS LXI    H,HORIZ*VERT+PAGE
E463 11B0FF                     LXI     D,-HORIZ
E466 3ADCFF                     LDA     LINENO
E469 3C                 CLOP    INR     A
E46A 19                         DAD     D
E46B FE18                       CPI     VERT
E46D 20FA                       JRNZ    CLOP            ;OPTIMIZED AT BOTTOM
E46F ED5BDBFF           CFIN    LDED    CURPOS
E473 1600                       MVI     D,0
E475 19                         DAD     D
E476                    * REVERSE THE VIDEO
E476 7E                         MOV     A,M
E477 EE80                       XRI     80H
E479 77                         MOV     M,A
E47A C9                         RET
E47B                    * CLEAR TO END OF SCREEN
E47B CD96E4             CLEND   CALL    WRSPC
E47E 18C2                       JR      RET
E480                    * CLEAR TO END OF LINE
E480 3ADBFF             CLLINE  LDA     CURPOS
E483 3620                       MVI     M,20H
E485 23                         INX     H
E486 3C                         INR     A
E487 FE50                       CPI     50H
E489 20F8                       JRNZ    CLLINE+3
E48B 18B5                       JR      RET
E48D                    * CLEAR THE SCREEN
E48D 2100F0             CLEAR   LXI     H,PAGE
E490 22DFFF                     SHLD    TOSCN
E493 22EAFF                     SHLD    XYFLAG
E496 3620               WRSPC   MVI     M,20H
E498 23                         INX     H
E499 7C                         MOV     A,H
E49A FEF8                       CPI     PAGE+2048/256
E49C 20F8                       JRNZ    WRSPC
E49E C9                         RET
E49F                    *
E49F                    * PROCESS LEAD IN CODE
```

```
E49F 3E02       LEDIN           MVI     A,2
E4A1 32EAFF                     STA     XYFLAG
E4A4 189C                       JR      RET
E4A6            *
E4A6            * SET X AND Y CURSOR POSITIONS
E4A6 79         XPOS            MOV     A,C
E4A7 FE50                       CPI     80
E4A9 3802                       JRC     XINRG
E4AB 3E4F                       MVI     A,79
E4AD 1890       XINRG           JR      TABRET
E4AF            *
E4AF 79         YPOS            MOV     A,C
E4B0 FE18                       CPI     24
E4B2 3802                       JRC     YINRG
E4B4 3E17                       MVI     A,23
E4B6 18A3       YINRG           JR      STORLN
E4B8            *
E4B8 AF         CLSTRT          XRA     A
E4B9 32DBFF                     STA     CURPOS
E4BC CD60E4                     CALL    LIFTCURS
E4BF 18BF                       JR      CLLINE
E4C1   E4C1 =   MSEND           EQU     $
E4C1            * CURSOR STORAGE LOCATIONS
E4C1                            ORG     SPTR+0BH
FFDB            CURPOS          DS      1               ;POS ON LINE
FFDC            LINENO          DS      1               ;LINE NUMBER
FFDD            VFL             DS      1               ;REVERSE VID FLAG
FFDE            WIDTH           DS      1               ;PRINT WIDTH
FFDF            TOSCN           DS      2               ;TOP OF SCREEN
FFE1            TCURPOS         DS      2               ;TEMP POSITION
FFE3                            LINK    'M5'
FFE3            * ADDITIONS TO 4.0 MONITOR
FFE3                            ORG     MSEND
E4C1            * PRINT A STRING
E4C1 CD0FE0     RPTSTNG         CALL    CRLF            ;CRLF FIRST
E4C4 E3         PTSTNG          XTHL
E4C5 7E                         MOV     A,M
E4C6 23                         INX     H
E4C7 E3                         XTHL
E4C8 A7                         ANA     A
E4C9 CD7BE3                     CALL    VIDEO           ;PRINT IT
E4CC F8                         RM
E4CD 18F5                       JR      PTSTNG
E4CF            * SIGN ON MESSAGE
E4CF 3E04       SIGN            MVI     A,4             ;CLEAR SCREEN
E4D1 CD7BE3                     CALL    VIDEO
E4D4 2150F1                     LXI     H,PAGE+150H
E4D7 E5                         PUSH    H
E4D8 1151F1                     LXI     D,PAGE+151H
E4DB 013000                     LXI     B,30H
E4DE 3612                       MVI     M,12H           ;GRAPHIC CHARACTER
E4E0 EDB0                       LDIR
E4E2 E1                         POP     H
E4E3 11A0F1                     LXI     D,PAGE+1A0H
E4E6 018002                     LXI     B,640
E4E9 EDB0                       LDIR
E4EB CDC4E4                     CALL    PTSTNG
```



```
E4EE 1B                         DB      27              ;ESC
E4EF 2007                       DB      2007H           ;X=32 Y=7
E4F1 20564543                   DT      ' VECTOR GRAPHIC '
E4F5 544F5220
E4F9 47524150
E4FD 48494320
E501 1B                         DB      27              ;ESC
E502 2008                       DB      2008H           ;X=32 Y=8
E504 20202020                   DT      '    MONITOR    '
E508 4D4F4E49
E50C 544F5220
E510 20202020
E514 1B                         DB      27              ;ESC
E515 2009                       DB      2009H           ;X=32 Y=9
E517 20205645                   DT      '  VERSION 4.2  '
E51B 5253494F
E51F 4E20342E
E523 32202020
E527 1B                         DB      27              ;ESC
E528 008D                       DB      80H             ;X=0 Y=13
E52A C9                         RET
E52B CDC1E4     PROMPT          CALL    RPTSTNG
E52E 4D6F6E3E                   DTH     'Mon> '
E532 A0
E533 C9                         RET
E534            *
E534            *WIDE ASCII DUMP
E534 CDC4E4     WASCII          CALL    PTSTNG
E537 41534349                   DTH     'ASCII DUMP '
E53B 49204455
E53F 4D50A0
E542 CD0EE1                     CALL    TAHEX
E545 CD88E5                     CALL    HOMEC
E548            * MAKE A RULER FOR ASCII DUMP
E548 78         RULELP          MOV     A,B
E549 FE40                       CPI     64
E54B 281A                       JRZ     TERMLIN
E54D E60F                       ANI     0FH
E54F 2810                       JRZ     NUMBER
E551 E603                       ANI     3
E553 2808                       JRZ     MARKER
E555 3E20                       MVI     A,' '
E557 CD7BE3     REENTR          CALL    VIDEO
E55A 04                         INR     B
E55B 18EB                       JR      RULELP
E55D 3E6C       MARKER          MVI     A,'|'
E55F 18F6                       JR      REENTR
E561 78         NUMBER          MOV     A,B
E562 CD2DE2                     CALL    BINH
E565 18F3                       JR      REENTR+3
E567            * TOGGLE REVERSE VIDEO
E567 CD79E3     TERMLIN         CALL    TVIDEO
E56A CDF4E5     WDMP1           CALL    SETSCRLL
E56D CD0FE2                     CALL    PTAD
E570 0E3F                       MVI     C,63
E572 CD79E5                     CALL    WDMP2
E575 FA6AE5                     JM      WDMP1
```

```
E578 C8                         RZ
E579 7E             WDMP2       MOV     A,M
E57A 47                         MOV     B,A
E57B 3E05                       MVI     A,'E'-64
E57D CD70E3                     CALL    VIDEO
E580 CD3FE2                     CALL    BMP
E583 C8                         RZ
E584 0D                         DCR     C
E585 F8                         RM
E586 18F1                       JR      WDMP2
E588                * HOME CURSOR, PRINT "ADDR"
E588 CDC1E4         HOMEC       CALL    RPTSTNG
E58B 14                         DB      'T'-64
E58C 41444452                   DTH     'ADDR '
E590 A0
E591 0600                       MVI     B,0
E593 3E18                       MVI     A,24
E595 32DEFF                     STA     WIDTH
E598 C9                         RET
E599                * MAKE A RULER FOR HEX DUMP
E599 78             HEXRULER    MOV     A,B
E59A FE10                       CPI     16
E59C 2806                       JRZ     HEXRCT
E59E CD1CE7                     CALL    PT2S
E5A1 04                         INR     B
E5A2 18F5                       JR      HEXRULER
E5A4                * EXTEND FOR ASCII
E5A4 CDDAE0         HEXRCT      CALL    SPCE
E5A7 CDDAE0                     CALL    SPCE
E5AA 0600                       MVI     B,0
E5AC 78             HEXRLP      MOV     A,B
E5AD FE10                       CPI     16
E5AF C8                         RZ
E5B0 E60F                       ANI     0FH
E5B2 CD31E2                     CALL    BINL
E5B5 04                         INR     B
E5B6 18F4                       JR      HEXRLP
E5B8                * HEX DUMP ROUTINE
E5B8 CDC4E4         HEXRUL      CALL    PTSTNG
E5BB 48455820                   DTH     'HEX DUMP '
E5BF 44554D50
E5C3 A0
E5C4 CD0EE1                     CALL    TWHEX
E5C7 CD88E5                     CALL    HOMEC
E5CA CD99E5                     CALL    HEXRULER
E5CD CD79E3                     CALL    TVIDEO
E5D0 CDF4E5                     CALL    SETSCRLL
E5D3 CD0FE2         HLP1        CALL    PTAD
E5D6 E5                         PUSH    H
E5D7 D5                         PUSH    D
E5D8 0E10                       MVI     C,16
E5DA 7E             HLP2        MOV     A,M
E5DB CD1CE7                     CALL    PT2S
E5DE 23                         INX     H
E5DF 0D                         DCR     C
E5E0 C2DAE5                     JNZ     HLP2
E5E3 D1                         POP     D
```



```
E5E4 E1                         POP     H
E5E5 0E0F                       MVI     C,15
E5E7 CDDAE0                     CALL    SPCE
E5EA CDDAE0                     CALL    SPCE
E5ED CD79E5                     CALL    WDMP2
E5F0 FAD0E5                     JM      HLP1-3
E5F3 C9                         RET
E5F4                * CHECK TO SET SCROLL POINT
E5F4 3ADEFF         SETSCRLL    LDA     WIDTH
E5F7 3D                         DCR     A
E5F8 32DEFF                     STA     WIDTH
E5FB 2007                       JRNZ    CTSCRL
E5FD 0150F0                     LXI     B,PAGE+50H      ;2ND LINE
E600 ED43DFFF                   SBCD    TOSCN           ;SCROLL POINT
E604 C9             CTSCRL      RET
E605                *
E605                * PROGRAM MEMORY
E605 CDC4E4         PROGRAM     CALL    PTSTNG
E608 50524F47                   DTH     'PROGRAM '
E60C 52414DA0
E610 CD8DE0                     CALL    AHEX            ;ADDR IN HL
E613 ED53E1FF                   SDED    TCURPOS
E617 CD88E5                     CALL    HOMEC           ;PRINT "ADDR"
E61A CD99E5                     CALL    HEXRULER
E61D CD79E3                     CALL    TVIDEO
E620 AF                         XRA     A
E621 32DEFF                     STA     WIDTH
E624 CD8EE6                     CALL    PRTLINE         ;PRINT LINE CONT H
E627 CD2FE1         POLLOOP     CALL    ESCAPE
E62A CDEDE0                     CALL    HEX
E62D 2AE1FF                     LHLD    TCURPOS
E630 301A                       JRNC    MODMEM
E632                * CONTROL CODE TABLE
E632 FE20                       CPI     ' '
E634 2846                       JRZ     CSRT
E636 FE08                       CPI     8
E638 2845                       JRZ     CSLT
E63A FE12                       CPI     'R'-64
E63C 2839                       JRZ     CSDN
E63E FE15                       CPI     'U'-64
E640 282F                       JRZ     CSUP
E642 FE17                       CPI     'W'-64
E644 2839                       JRZ     CSLT
E646 FE1A                       CPI     'Z'-64
E648 2832                       JRZ     CSRT
E64A 18DB                       JR      POLLOOP
E64C                * MODIFY A MEMORY LOCATION
E64C 2AE1FF         MODMEM      LHLD    TCURPOS
E64F 4F                         MOV     C,A
E650 3ADEFF                     LDA     WIDTH
E653 A7                         ANA     A
E654 7E                         MOV     A,M
E655 280D                       JRZ     LSNIBL
E657 E6F0                       ANI     0F0H
E659 B1                         ORA     C
E65A 77             REMEM       MOV     M,A
E65B 3ADEFF                     LDA     WIDTH
```

```
E65E EE01                       XRI     1
E660 201F                       JRNZ    RTRTN+1
E662 1818                       JR      CSRT
E664 17         LSNIBL          RAL
E665 17                         RAL
E666 17                         RAL
E667 17                         RAL
E668 E6F0                       ANI     0F0H
E66A B1                         ORA     C
E66B 0F                         RRC
E66C 0F                         RRC
E66D 0F                         RRC
E66E 0F                         RRC
E66F 18E9                       JR      REMEM
E671            * MOVE UP ONE LINE
E671 11F0FF     CSUP            LXI     D,-16
E674 19                         DAD     D
E675 1809                       JR      RTRTN
E677            * MOVE DOWN ONE LINE
E677 111000     CSDN            LXI     D,16
E67A 18F8                       JR      CSUP+3
E67C            * MOVE RIGHT ONE SPACE
E67C 23         CSRT            INX     H
E67D 1801                       JR      RTRTN
E67F            * MOVE LEFT ONE SPACE
E67F 2B         CSLT            DCX     H
E680            *
E680 AF         RTRTN           XRA     A
E681 32DEFF                     STA     WIDTH
E684 22E1FF                     SHLD    TCURPOS
E687 3E15       UPAROW          MVI     A,'U'-64
E689 CD7BE3                     CALL    VIDEO
E68C 1896                       JR      POLLOOP-3
E68E            * PRINT A LINE CONTAINING ((H))
E68E 2AE1FF     PRTLINE         LHLD    TCURPOS
E691 E5                         PUSH    H
E692 D1                         POP     D
E693 7D                         MOV     A,L
E694 F60F                       ORI     0FH
E696 5F                         MOV     E,A
E697 E6F0                       ANI     0F0H
E699 6F                         MOV     L,A
E69A CDD3E5                     CALL    HLP1
E69D            * NOW PUT CURSOR WHERE IT GOES
E69D CD60E4                     CALL    LIFTCURS
E6A0 2AE1FF                     LHLD    TCURPOS
E6A3 7D                         MOV     A,L
E6A4 E60F                       ANI     0FH
E6A6 6F                         MOV     L,A
E6A7 3E05                       MVI     A,5
E6A9 2D         PLOP1           DCR     L
E6AA FAB1E6                     JM      PGCONT
E6AD C603                       ADI     3
E6AF 18F8                       JR      PLOP1
E6B1 6F         PGCONT          MOV     L,A
E6B2 3ADEFF                     LDA     WIDTH
E6B5 85                         ADD     L
```

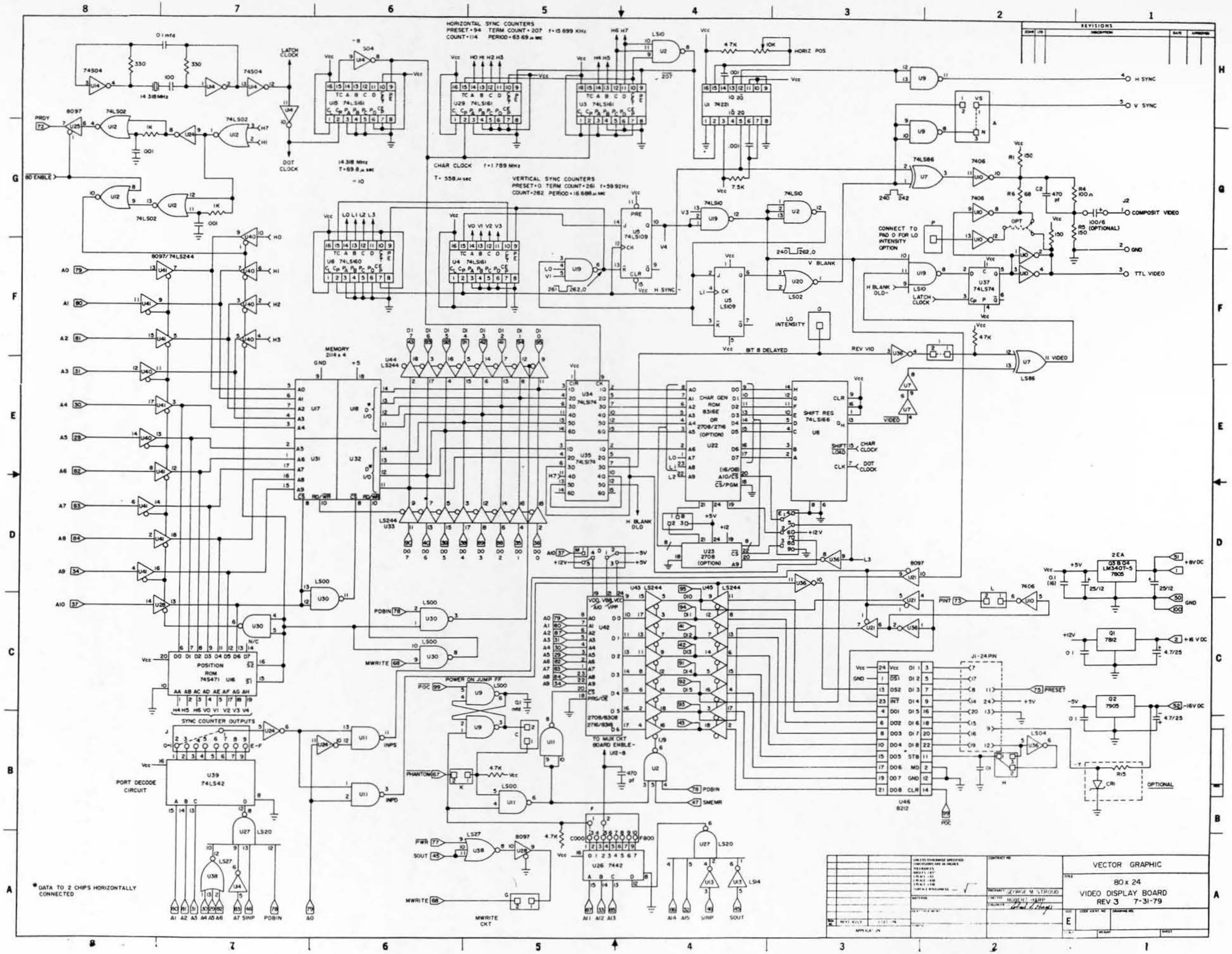


```
E6B6            * A = 5+3*L+W
E6B6 32D0FF                     STA     CURPOS
E6B9 C360E4                     JMP     LIFTCURS
E6BC            *
E6BC            *
E6BC            * DISPLAY REGISTERS
E6BC CDC4E4     DREGS           CALL    PTSTNG
E6BF 52454749                   DTH     'REGISTERS'
E6C3 53544552
E6C7 D3
E6C8            * DUMP REGISTERS AFTER ENTRY FROM RST 7
E6C8 E3         DUMPREGS        XTHL
E6C9 F5                         PUSH    PSW
E6CA CD22E7                     CALL    DISPREGS
E6CD 2B                         DCX     H               ;GET BREAK ADD
E6CE CD0FE2                     CALL    PTAD
E6D1 E1                         POP     H
E6D2 C5                         PUSH    B
E6D3 CD77E7                     CALL    PRTFLGS
E6D6 C1                         POP     B
E6D7 CD12E2                     CALL    PTAD+3          ;PRINT AF
E6DA E1                         POP     H
E6DB 22E3FF                     SHLD    HLTEMP
E6DE CD98E7                     CALL    PTHREE          ;PRINT B D H
E6E1 DDE5                       PUSH    IX
E6E3 E1                         POP     H
E6E4 CD12E2                     CALL    PTAD+3          ;PRINT IX
E6E7 FDE5                       PUSH    IY
E6E9 E1                         POP     H
E6EA CD12E2                     CALL    PTAD+3          ;PRINT IY
E6ED 210000                     LXI     H,0
E6F0 39                         DAD     SP
E6F1 22E5FF                     SHLD    SPTEMP
E6F4 CD12E2                     CALL    PTAD+3          ;PRINT SP
E6F7 08                         EXAF
E6F8 F5                         PUSH    PSW
E6F9 E1                         POP     H
E6FA CD12E2                     CALL    PTAD+3
E6FD D9                         EXX
E6FE CD98E7                     CALL    PTHREE
E701 D9                         EXX
E702 0A                         LDAX    B
E703 CD1CE7                     CALL    PT2S
E706 1A                         LDAX    D
E707 CD1CE7                     CALL    PT2S
E70A 2AE3FF                     LHLD    HLTEMP
E70D 7E                         MOV     A,M
E70E CD1CE7                     CALL    PT2S
E711 2AE5FF                     LHLD    SPTEMP
E714 F9                         SPHL
E715 E1                         POP     H
E716 CD12E2                     CALL    PTAD+3
E719 C340E0                     JMP     CLRBRK          ;CLEAR BREAKPOINT
E71C            *
E71C CD26E2     PT2S            CALL    PT2             ;PRINT 2 CHARS
E71F C3DAE0                     JMP     SPCE            ;PRINT SPACE
E722            * DISPLAY REGISTER HEADER ON SCREEN
```

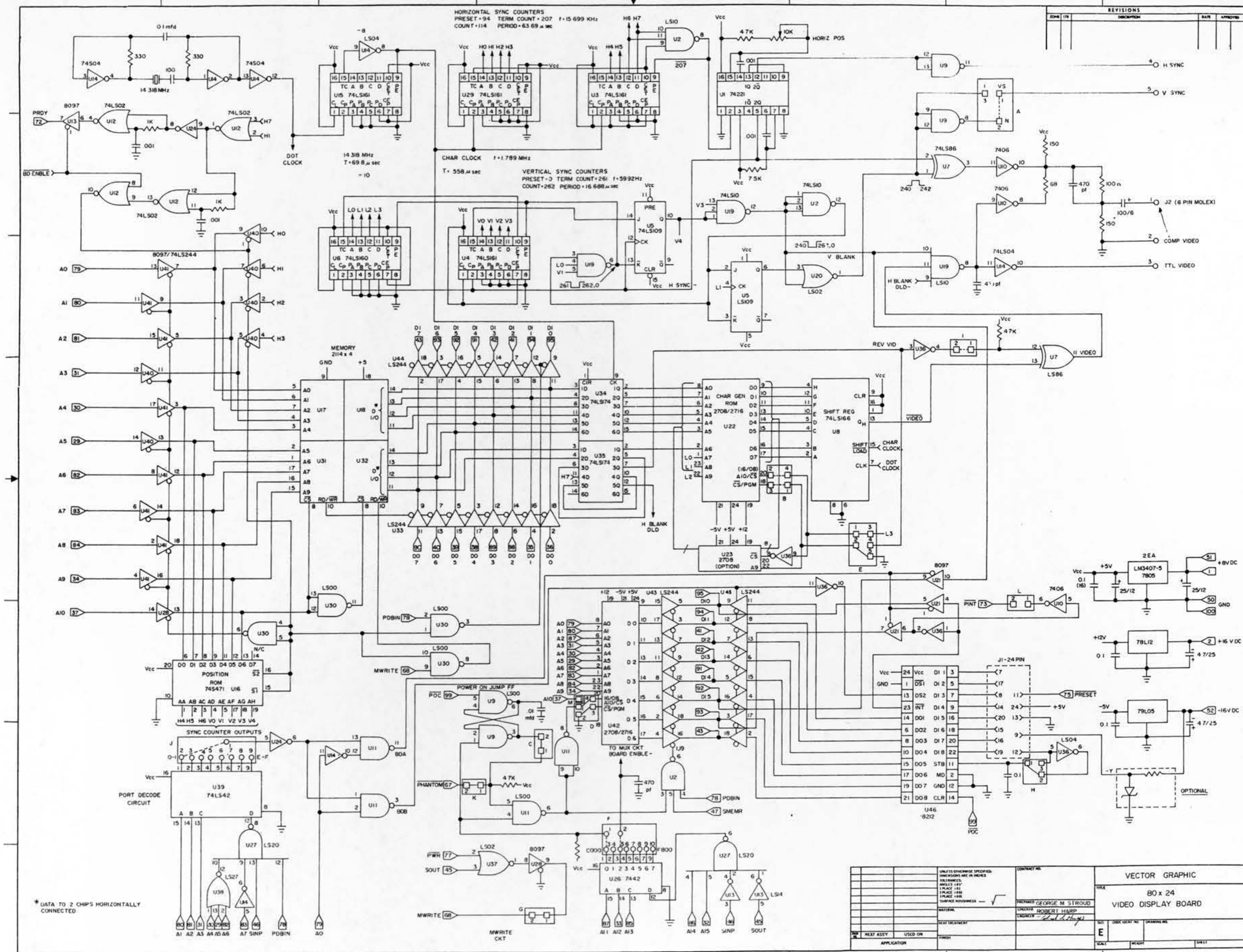
```
E722 CDC1E4     DISPREGS        CALL    RPTSTNG
E725 14                         DB      'T'-64      *
E726 41444452                   DT      'ADDR FLAGS  AF   BC   DE'
E72A 20464C41
E72E 47532020
E732 41462020
E736 20424320
E73A 20204445
E73E 20202048                   DT      '   HL   IX   IY   SP '
E742 4C202020
E746 49582020
E74A 20495920
E74E 20205350
E752 20
E753 20204146                   DT      '  AF'
E757 27                         DB      27H             ;'
E758 20204243                   DT      '  BC'
E75C 27                         DB      27H
E75D 20204445                   DT      '  DE'
E761 27                         DB      27H
E762 2020484C                   DT      '  HL'
E766 27                         DB      27H
E767 20404220                   DT      ' @B @D @H @SP '
E76B 40442040
E76F 48204053
E773 5020
E775 94                         DB      'T'+64
E776 C9                         RET
E777            *
E777            * PRINT FLAGS
E777 015A40     PRTFLGS         LXI     B,405AH         ;Z
E77A CDA7E7                     CALL    MASKFLG
E77D 014301                     LXI     B,143H          ;C
E780 CDA7E7                     CALL    MASKFLG
E783 014D80                     LXI     B,804DH         ;M
E786 CDA7E7                     CALL    MASKFLG
E789 014504                     LXI     B,445H          ;E
E78C CDA7E7                     CALL    MASKFLG
E78F 014810                     LXI     B,1048H         ;H
E792 CDA7E7                     CALL    MASKFLG
E795 C3DAE0                     JMP     SPCE
E798            *
E798            * PRINT BC DE HL IN ORDER
E798 E5         PTHREE          PUSH    H
E799 C5                         PUSH    B
E79A E1                         POP     H
E79B CD12E2                     CALL    PTAD+3
E79E D5                         PUSH    D
E79F E1                         POP     H
E7A0 CD12E2                     CALL    PTAD+3
E7A3 E1                         POP     H
E7A4 C312E2                     JMP     PTAD+3
E7A7            *
E7A7 7D         MASKFLG         MOV     A,L
E7A8 A0                         ANA     B
E7A9 3E20                       MVI     A,20H
E7AB CA78E3                     JZ      VIDEO
```



```
E7AE 79                         MOV     A,C
E7AF C37BE3                     JMP     VIDEO
E7B2            *
E7B2            * SET BREAKPOINT
E7B2 CDC4E4     SETBRK          CALL    PTSTNG
E7B5 42524541                   DTH     'BREAK AT '
E7B9 4B204154
E7BD A0
E7BE CDBDE0                     CALL    ANEX
E7C1 1A                         LDAX    D
E7C2 32E9FF                     STA     BRKCODE
E7C5 ED53E7FF                   SDED    BKPTLOC
E7C9 3EFF                       MVI     A,0FFH          ;RST 7
E7CB 12                         STAX    D
E7CC C9                         RET
E7CD            *
E7CD            * EXTERNAL COMMUNICATIONS
E7CD CDC4E4     EXTCOM          CALL    PTSTNG
E7D0 45585420                   DTH     'EXT COM '
E7D4 434F4DA0
E7D8 DB05       RECEIVE         IN      5
E7DA E602                       ANI     2
E7DC 2805                       JRZ     NEXCHR
E7DE DB04                       IN      4
E7E0 CD7BE3                     CALL    VIDEO
E7E3 CD2FE1     NEXCHR          CALL    ESCAPE
E7E6 28F0                       JRZ     RECEIVE
E7E8 D304                       OUT     4
E7EA 18EC                       JR      RECEIVE
E7EC            *
E7EC            * TEMPORARY STORAGE LOCATIONS FOR REGISTERS,ETC.
E7EC                            ORG     TOURPOS+2
FFE3            HLTEMP          DS      2
FFE5            SPTEMP          DS      2
FFE7            BKPTLOC         DS      2               ;BREAKPT LOCATION
FFE9            BRKCODE         DS      1               ;CODE AT BREAKPT
FFEA            XYFLAG          DS      1               ;CURSOR XY FLAG
```



HORIZONTAL SYNC COUNTERS
PRESET = 94 TERM COUNT = 207 f = 15.699 KHz
COUNT = 114 PERIOD = 63.69 μ sec
VERTICAL SYNC COUNTERS
PRESET = 0 TERM COUNT = 261 f = 59.92 Hz
COUNT = 262 PERIOD = 16.688 μ sec
CHAR CLOCK f = 1.789 MHz
14.318 MHz
LATCH CLOCK
DOT CLOCK
HORIZ POS
H SYNC
V SYNC
COMPOSIT VIDEO
GND
TTL VIDEO
CONNECT TO PAD 0 FOR LO INTENSITY OPTION
LO INTENSITY
V BLANK
H BLANK OLD
BIT 8 DELAYED
REV VID
VIDEO
MEMORY 2114 x 4
CHAR GEN ROM 8316E OR 2708/2716 (OPTION) U22
SHIFT REG 74LS166
POSITION ROM 74S471 U16
SYNC COUNTER OUTPUTS
PORT DECODE CIRCUIT
U39 74LS42
POWER ON JUMP FF
TO MUX CKT BOARD ENBLE
MWRITE CKT
J1 - 24 PIN
U46 8212
PRESET
OPTIONAL
+8V DC
+16 V DC
-16V DC
* DATA TO 2 CHIPS HORIZONTALLY CONNECTED
VECTOR GRAPHIC
80 x 24
VIDEO DISPLAY BOARD
REV 3 7-31-79


HORIZONTAL SYNC COUNTERS
PRESET = 94 TERM COUNT = 207 f = 15.699 KHz
COUNT = 114 PERIOD = 63.69 μ sec
VERTICAL SYNC COUNTERS
PRESET = 0 TERM COUNT = 261 f = 59.92 Hz
COUNT = 262 PERIOD = 16.688 μ sec
CHAR CLOCK f = 1.789 MHz
T = 558 μ sec
14.318 MHz
T = 69.8 μ sec
÷ 10
DOT CLOCK
HORIZ POS
H SYNC
V SYNC
J2 (6 PIN MOLEX)
COMP VIDEO
TTL VIDEO
V BLANK
H BLANK DLD
REV VID
VIDEO
MEMORY 2114 x 4
CHAR GEN ROM 2708/2716
SHIFT REG 74LS166
POSITION ROM 74S471
SYNC COUNTER OUTPUTS
PORT DECODE CIRCUIT
POWER ON JUMP FF
TO MUX CKT BOARD ENBLE
MWRITE CKT
LM3407-5 7805
78L12
79L05
+8V DC
+16 V DC
-16V DC
GND
OPTIONAL
J1-24 PIN
PRESET
* DATA TO 2 CHIPS HORIZONTALLY CONNECTED
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN INCHES
NEXT ASS'Y
USED ON
APPLICATION
MATERIAL
HEAT TREATMENT
FINISH
CONTRACT NO.
PREPARED GEORGE M STROUD
CHECKED ROBERT HARP
ENGINEER
VECTOR GRAPHIC
TITLE
80 x 24
VIDEO DISPLAY BOARD
SIZE E
CODE IDENT NO
DRAWING NO.
SCALE
WEIGHT
SHEET
REVISIONS