VB3
80 CHARACTER VIDEO INTERFACE
S-100 BUS

INSTRUCTION MANUAL

SEPTEMBER 1981

SSM Microcomputer Products, Inc.
2190 Paragon Drive
San Jose, California 95131

TELEPHONE: (408) 946-7400
TWX: 910-338-2077
TELEX: 171171
DDD: (408) 946-3644 (110 BAUD)

VB3
80 CHARACTER VIDEO INTERFACE
S-100 BUS

INSTRUCTION MANUAL

SEPTEMBER 1981

SSM Microcomputer Products, Inc.
2190 Paragon Drive
San Jose, California 95131

TELEPHONE: (408) 946-7400
TWX: 910-338-2077
TELEX: 171171
DDD: (408) 946-3644 (110 BAUD)

IMPORTANT

MODIFICATION

To Your VB3 Kit

Congratulations on buying your SSM VB3 kit, and welcome to the group of thousands of satisfied SSM customers. We were unable to incorporate the latest engineering changes into the original photographic artwork and still manufacture the boards within our delivery schedule commitments. Therefore, we have cut 5 traces, and would like you to add 4 short jumper wires to your board after you have completed construction. The correct schematic is included in your manual. Please use the instructions on the reverse side of this sheet. We want to assure you that your board still has its full SSM warranty, and we stand behind it as we do for all of our products. We're sure you will be pleased with its performance.

5/22/80
Rev. 1

VB3 KIT
MODIFICATION INSTRUCTIONS

NOTE:

- [x] Silkscreen for "C11" polarity is missing. Install "C11" with "+" sign to the right (looking at component side of board, with bus connector fingers down).

- [x] Silkscreen for "CR1" polarity is missing. Install "CR1" with band to the right (looking at component side of board, with bus connector fingers down).

Ater you have performed all of the assembly instructions, then:

- [x] Cut trace, on non-component side of board, from U12, pin 9 to U12, pin 10. (We did it for you.)

- [x] Cut trace, on non-component side of board, from U12, pin 8 to U1, pin 12. (Don't worry about tracking the trace all the way to U1, pin 12. There is only one trace coming off of U12, pin 8.) (We did it for you.)

- [x] On non-component side of board, add jumper wire (supplied) from U12, pin 8, to U12, pin 9.

- [x] On non-component side of board, add insulated jumper wire (supplied) from U12, pin 10 to U1, pin 12.

- [x] Cut trace, on non-component side of board, between U1, pin 1 and U2, pin 20 (etc.). Cut the trace near U1, pin 1 before the feedthrough eyelet. (We did it for you.)

- [x] Cut trace, on non-component side of board, between U1, pin 2 and the nearby feedthrough eyelet (after the "L" shaped junction of the trace that goes to U2, pin 6). In other words, cut the portion of the trace between the feedthrough eyelet and the trace that connects U1, pin 2 and U2, pin 6. (We did it for you.)

- [x] Cut trace, on non-component side of board, (that was) between U1, pin 1 and U2, pin 10. Cut it fairly close to U2, pin 10. (We did it for you.)

- [x] On non-component side of board, add insulated jumper wire (supplied) from U1, pin 2 to U23, pin 3.

- [x] On non-component side of board, add insulated jumper wire (supplied) from U1, pin 1 to U23, pin 1.

- [x] On non-component side of board, add insulated jumper wire (supplied) from U1, pin 1 to U2, pin 10.

- [x] FINISHED! (Enjoy)

TABLE OF CONTENTS

Edited By: Dan Fischler
Illustrated By: Judith Sisko

Part Number: MN0023

1.0 INTRODUCTION

The SSM VB3 memory mapped video display board provides a flexible video display system for S-100 bus computers.

A maximum of 4096 bytes of contiguous memory may be directly mapped to the screen as characters or graphics.

The display may be programmed for up to fifty 80-character lines (or fifty-one on European standard monitors) featuring upper and lower case letters with descenders. Optionally, the user may display 20, 32, 40, 64, 72, 96, and 132-characters per line using optional mapping PROM's available at extra cost from SSM.

The VB3 features a second RAM block in addition to the video RAM which contains "attribute" bytes to control the display of each individual character. These attributes allow any individual character to appear as a standard alphanumeric upper/lower case font or an alternate user-programmed font. (SSM includes with the VB3 one alternate character font. This is a 6 x 7 matrix character set for displaying the maximum number of text lines using the lowest number of raster lines possible.) In addition, the character may be displayed in normal or low intensity, reverse video (black on white), with an underscore or strike-through mark, blinking, blanked or as a thin line or dot graphics.

2.0 ASSEMBLY INSTRUCTIONS

Refer to the ASSEMBLY DIAGRAM and JUMPER DIAGRAM in the Appendix during assembly and test procedures.

2.1 UNPACKING

[ ] Unpack and check each of the parts against the PARTS LIST provided.

[ ] It is a good idea to arrange the parts in a small tray or box to allow for easy identification and accessibility during assembly.

2.2 RESISTOR INSTALLATION

NOTE: Be sure all resistors and diodes are flush against the PC board. This will insure proper socket installation. **DO NOT** install R19 at this time.

[ ] Install and solder THREE (3) 100 ohm resistors (brown, black, brown) at locations R5, 6, and 10.

[ ] Install and solder TWO (2) 1K ohm resistors (brown, black, red) at locations R8 and R9.

[ ] Install and solder FIFTEEN (15) 10K ohm resistors (brown, black, orange) at locations R1, 2, 3, 4, 13, 20, 21, 22, 23, 24, 25, 26, 28, 29, and 30.

[ ] Install and solder TWO (2) 220 ohm resistors (red, red, brown) at locations R7 and R27.

[ ] Install and solder ONE (1) 390 ohm resistor (orange, white, brown) at location R11.

[ ] Install and solder THREE (3) 2.7K ohm resistors (red, violet, red) at locations R14, R15, and R18.

[ ] Install and solder TWO (2) 470 ohm resistors (yellow, violet, brown) at locations R16 and R17.

[ ] Install and solder ONE (1) 20K ohm resistor (red, black, orange) at location R12.

[ ] Install and solder THREE (3) 2.7K ohm resistor networks at locations RP1, 2, and 3. Use caution in installing these components-- RP1 MUST have Pin 1 to the LEFT side of the board. RP2 and RP3 MUST have Pin 1 towards the TOP of the board.

2.3 DIODE INSTALLATION

[ ] Install and solder ONE (1) 1N5242 (alternately a 1N4742) zener diode at location CR1. Use caution in installing this component-- the banded end (+) MUST be to the RIGHT side of the board.

2.4 SOCKET INSTALLATION

NOTE: DO NOT install integrated circuits until specifically instructed to do so.

[ ] Install the 14, 16, 18, 24, 28, and 40 pin IC sockets on the printed circuit board. Orient Pin 1 towards the top of the board or to the left, as applicable. See Figure 1 for information on locating Pin 1 on each socket.

NOTE: DO NOT install sockets at locations S1, S2, S3, or S4. Switches will be installed at these locations in a later step.

FOURTEEN (14)	14-pin sockets at U6,8,9,10,12,13,14, 15,19,20,keyboard, 24,30,35
TWENTY-ONE (21)	16-pin sockets at U2,3,4,5,7,11,16,17, 18,21,22,23,25,26,29, 32,33,34,36,37,38
SIXTEEN (16)	18-pin sockets at U39,40,41,42,43,44,45, 46,47,48,49,50,51,52, 53,54
TWO (2)	24-pin sockets at U27,28
ONE (1)	28-pin socket at U31
ONE (1)	40-pin socket at U1

SOCKET TYPES

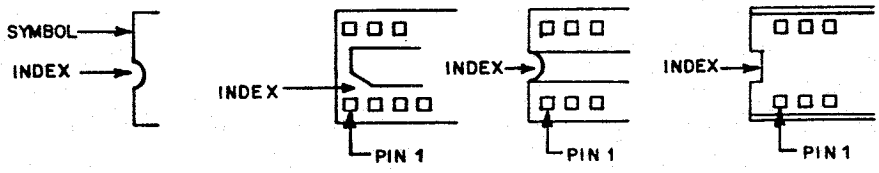


FIGURE 1

[ ] When all sockets are inserted, place a piece of stiff cardboard over the sockets to hold them in place and turn the board over to expose the reverse side.

[ ] On each socket, solder Pin 1 and the pin diagonally opposite it to 'tack' (lightly solder) each socket in place. When all sockets are tacked in place, turn the board over and examine each socket to make sure it is flush against the board. If needed, reheat the pins and adjust any sockets not firmly mounted.

[ ] When all sockets are properly seated, solder the remaining pins of each socket. **DO NOT** overheat.

2.5 CAPACITOR INSTALLATION

NOTE: DO NOT install C16 at this time.

[ ] Install and solder TWENTY-NINE (29) 0.1 uf monolithic capacitors at locations C1, 2, 3, 4, 5, 7, 8, 12, 13, 14, 15, 17, 18, 19, 20, 21, 22, 24, 25, 26, 27, 28, 29, 31, 32, 33, 34, 35, and 36.

[ ] Install and solder ONE (1) 100 pf disc ceramic capacitor at location C6.

[ ] Install and solder ONE (1) 1000 pf (.001 uf) disc ceramic capacitor at location C9.

[ ] Install and solder FOUR (4) 4.7 uf dipped tantalum capacitors at locations C10, 11, 23, and 30. Use caution in installing these components-- C10 and C23 MUST have the positive (+) end to the LEFT side of the board, C11 MUST have the positive (+) end to the RIGHT side of the board, and C30 MUST have the positive (+) end towards the BOTTOM of the board.

2.6 TRANSISTOR INSTALLATION

[ ] Install and solder ONE (1) 2N3904 (alternately a 2N2222) transistor at location Q1. Align the leads with the respective holes, align the case with the silkscreen, and insert so that about 1/4 inch of lead remains between the body of the transistor and the component side of the board.

2.7 CRYSTAL INSTALLATION

[ ] Install and solder ONE (1) 16 MHz crystal at location Y1. Two holes have been provided on either side of the crystal to solder a strap over the crystal to hold it down. Use a resistor lead to make this strap. **DO NOT** overheat the crystal.

2.8 REGULATOR INSTALLATION

[ ] Place FOUR (4) 7805 regulators on the board so that the mounting hole in the regulator **is** in line with the hole in the board. Mark the leads for proper bending to match the holes in the board (allow for bend radius).

[ ] Bend the regulator leads to match the holes in the board.

[ ] If available, apply thermal compound to the back side of each regulator case (the side that will contact the heatsink). Use just a little thermal compound. **Too much is worse than none at all.**

[ ] Install and solder THREE (3) 7805 regulators at locations X1, 2, and 3 so that the following order results from back to front: screw, PC board, heatsink, regulator, lock washer, and nut. Be sure the regulators and heatsinks lay flush against the board and then solder all regulator leads.

[ ] Install and solder ONE (1) 7805 regulator at location X4 using the special heatsink provided (This heatsink has a double-hole drilled in it. Refer to Figure 2 for further information) so that the following order results from back to front: screw, PC board, heatsink, regulator, lock washer, and nut. Be sure the regulator and heatsink lay flush against the board and then solder all regulator leads.

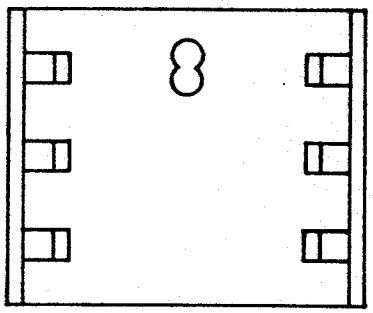


FIGURE 2

[ ] Install and solder ONE (1) 78L12AC regulator at location Q2. Align the leads with the respective holes and insert so that about 1/4 inch of lead remains between the body of the regulator and the component side of the board.

2.9 CONNECTOR, HEADER, and SWITCH INSTALLATION

[ ] Install and solder the 20 pin right-angle connector at the upper left corner of the board. Position the connector such that the side with the 90 degree bend is soldered to the PC board. See Figure 3 for proper orientation.

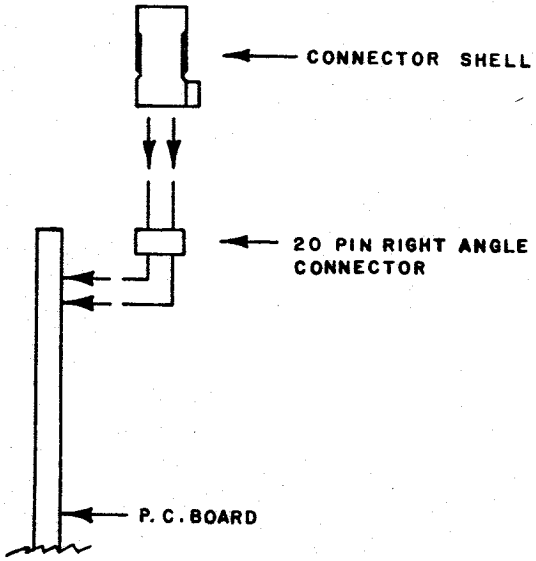


FIGURE 3

[ ] Install and solder TWO (2) sets of three wire-wrap header pins at locations E2, 3, 4 and E5, 6, 7 on the front of the board (short pin end into board). Cut to length.

[ ] Install and solder TWO (2) four position DIP switches at locations S1 and S2. Be sure position 1 is to the left.

[ ] Install and solder TWO (2) eight position DIP switches at locations S3 and S4. Be sure position 1 is toward the top of the board.

3.0 FUNCTIONAL CHECK

WARNING! DO NOT INSTALL OR REMOVE BOARDS WITH POWER ON. DAMAGE TO THIS AND OTHER BOARDS COULD RESULT.

3.1 SHORT TEST

[ ] If an ohmmeter is available, measure the resistance between pin 50 (negative meter probe) and pin 1 (positive meter probe) on the edge connector and verify a resistance of 20 ohms or greater. If your reading is below 20 ohms, check your board for possible shorts.

[ ] If an ohmmeter is available, measure the resistance between pin 50 (negative meter probe) and pin 2 (positive meter probe) on the edge connnector and verify a resistance of 1000 ohms or greater. If your reading is below 1000 ohms, check your board for possible shorts.

[ ] If an ohmmeter is available, measure the resistance between pin 52 (negative meter probe) and pin 50 (positive meter probe) on the edge connector and verify a resistance of 1000 ohms or greater. If your reading is below 1000 ohms, check your board for possible shorts.

3.2 VOLTAGE CHECK

[ ] Apply power (+8v to +10v) to the board by plugging into the computer or by connection to a suitable power supply (with power turned-off). Measure the outputs of the +5v regulators (X1, X2, X3, X4). The voltage should be +5.0v (+/- 0.2v). If the regulator doesn't meet this test, check the board for shorts or errors. (See Figure 4 for the pin assignments of the regulator.)

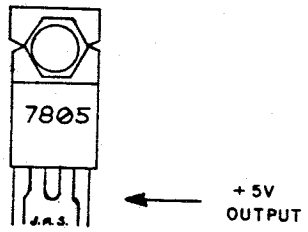


FIGURE 4

CAUTION: WHILE IT HAS NEVER HAPPENED TO US, SHORTED REGULATORS HAVE BEEN KNOWN TO EXPLODE WITH POSSIBLE INJURY TO EYES AND HANDS. BETTER SAFE THAN SORRY-- KEEP YOUR FACE AND HANDS CLEAR OF THE REGULATOR SIDE OF THE BOARD DURING THE INITIAL POWER-UP OF YOUR BOARD.

[ ] Apply power (+16v to +18v) to the board by plugging into the computer or by connection to a suitable power supply (with power turned-off). Connect the negative meter probe to U1 pin 6 and the positive meter probe to U1 pin 13. The voltage present should be +12v (+/- 0.6v). If the regulator doesn't meet this test, check the board for shorts or opens to Q2.

[ ] Apply power (-16v to -18v) to the board by plugging into the computer or by connection to a suitable power supply (with power turned-off). Connect the negative meter probe to point E1 (refer to the JUMPER DRAWING in the APPENDIX for location) and the positive meter probe to the cathode of CR1 (the banded end). The voltage present should be -12v (+/- 0.6v). If the voltage present is not within this range, check the board for shorts or errors.

3.3 VISUAL INSPECTION

[ ] Now, look over the board carefully. Check for solder bridges, colder solder joints, and unsoldered pins. Also, using the ASSEMBLY DRAWING in the APPENDIX, check for improper part location or polarity. A few minutes of careful inspection could save hours in troubleshooting later.

3.4 IC INSTALLATION/TEST PROCEDURE

[ ] Refer to the ASSEMBLY DIAGRAM in the APPENDIX to install the following integrated circuits. BE CERTAIN THAT PIN 1 OF EACH IC IS ORIENTED PROPERLY. It is sometimes helpful to bend the leads of the IC's SLIGHTLY inward by placing the IC package on its side and applying firm pressure. This assures the leads will be straight and makes it easier to install the device in a socket.

3.41 MEMORY TEST

[ ] Install the following IC's as shown in the ASSEMBLY DIAGRAM:

[ ] U2	74LS257
[ ] U5	74LS139
[ ] U6	74LS86
[ ] U7	74LS163
[ ] U8	74LS74
[ ] U9	74S02
[ ] U10	7406/7416
[ ] U12	74LS02
[ ] U13	74LS08
[ ] U14,U35	74LS32
[ ] U15,U20	74LS04
[ ] U16	74LS175
[ ] U21,U22,U29	74LS367
[ ] U25	DM8131
[ ] U30	74LS00
[ ] U33,U34,U37,U38	DM8216

[ ] At this point the VB3 is set up as a memory board only. Now insert the memory IC's. THE FOLLOWING IC'S ARE EXTREMELY SENSITIVE TO STATIC ELECTRICITY. AVOID TOUCHING THE IC LEADS WITHOUT FIRST TOUCHING THE BOARD TO MAKE SURE BOTH ITEMS ARE AT THE SAME STATIC POTENTIAL.

[ ] U39,40,43,44	2114L
[ ] U47,48,51,52	

[ ] The VB3 may now be tested as a standard 8K memory board (Refer to section 4.4 for proper address strobe and section 4.6 for memory address selection prior to running a memory test. The address strobe option **MUST** be selected to either 8080 or Z-80 mode prior to attempting this test. Failure to do so may inhibit normal system operation.). The VB3 memory map will start on an 8K boundary determined by the setting of switch S2 positions 1, 2, and 3 with 2K of RAM, 2K of 0FFH bytes, 2K of RAM, and 2K of 0FFH bytes. A memory test program has been provided in section 6.4 to perform this part of the VB3's check-out.

(If you have the 80 x 50 expansion option, add the memory chips at U41, 42, 45, 46, 49, 50, 53, 54 to make the VB3's memory contiguous for 8K).

3.42 KEYBOARD TEST

[ ] Install the following IC's as shown in the ASSEMBLY DRAWING:

[ ] U17	74LS367
[ ] U18	74LS139
[ ] U23	DM8131
[ ] U24	74LS136
[ ] U27	8212

[ ] The VB3's keyboard input may now be tested. Address the keyboard input to an unused I/O port pair. (See Section 4.3 for keyboard addressing). Temporarily ground pin 1 of the keyboard connector (keyboard socket pin 7 or 8) temporarily by a thin jumper wire, then the status port (KSTAT) will have bit D0=0 for testing purposes. Reading the data port (KDATA) will clear the D0 bit.

3.43 VIDEO TEST

[ ] Install the following IC's as shown in the ASSEMBLY DIAGRAM:

[ ] U3	82S131/DM54S571 (80-L)
[ ] U4	82S131/DM54S571 (80-H)
[ ] U11	74LS123/74123
[ ] U26	DM8131
[ ] U32,36	8216
[ ] U1	SMC5037
[ ] U31	SMC8002
[ ] U28	2716 (6 x 7 character set)

NOTE: Install U19 (74LS74) only after you have set up the board (Section 4.0) to your needs and operated it to verify no problems exist. U19 controls the bank select feature and need only be installed if the VB3 is going to overlay some other memory board equipped with Phantom Disable (Bus pin 67).

[ ] Enter the video test routine in section 6.5 into RAM starting at location 0100 hex. (This program is provided in two formats, 8080 assembly language and 8080 machine code.) When the program is fully entered, execute it at location 0100 (hex). The VB3 will be

initialized for an 80 x 16, non-interlaced display. This routine assumes the VB3's keyboard is addressed at port E0 (hex) and the CRT controller ports will start at D0 (hex).

4.0 SETTING UP YOUR VB3

4.1 EXTERNAL SYNC POLARITY

If external sync is not being used, all positions of Switch S1 should be left open (off). If external sync is being used, the switch settings depend on the polarity of the external sync signal.

SYNC POLARITY

> For a negative horizontal sync input (negative TTL pulses), set Switch S1, position 1 CLOSED (ON).

> For a positive horizontal sync input (positive TTL pulses), set Switch S1, position 1 OPEN (OFF).

> For a negative vertical sync input (negative TTL pulses, set Switch S1, position 2 CLOSED (ON).

> For a positive vertical sync input (positive TTL pulses), set Switch S1, position 2 OPEN (OFF).

The vertical and horizontal rates of the VB3 need to be set to a slightly higher frequency than the signal that you are trying to synchronize with, for correct operation. This can be done by picking a smaller 'HCOUNT' (see Section 5.1, Register 0 for information) to raise the vertical and horizontal rates proportionately.

4.2 CHARACTER SIZE (DOT WIDTH)

A character is sent out from the VB3 as a series of white dots at a rate of 16 MHz rate. (This rate is set by the frequency of crystal Y1. The frequency may need to be changed for special applications, for example, line lengths of other than 80 columns.) A counter is used on the VB3 to count the number of dots that will represent one character width, and feeds this character width internally to the CRT controller (U1, pin 12) to set up all the video timing.

The number of dots per character horizontally is set by Switch S2. The following settings are possible:

	SWITCH POSITION				
DOTS/CHARACTER	1	2	3	4	
6	1	0	1	0	0 = Closed
7	1	0	0	1	1 = Open
8	1	0	0	0	
9	0	1	1	1	
10	0	1	1	0	
11	0	1	0	1	
12	0	1	0	0	

NOTE: 9 dots per character is the recommended character width when using the SMC 8002 (7 x 9) character set. When using the alternate character set contained in U28 (6 x 7) it is recommended that 8 or 9 dots per character be used.

4.3 KEYBOARD ADDRESSING

Switch S3 is used to select the address of the keyboard status (KSTAT) and data port (KDATA) addresses.

The I/O ports may be set to any I/O port pair from 00 to FF (hex). Additionally, either the status or the data port may be set to the first I/O port of the pair.

KDATA PORT ADDRESS	1 A7	2 A6	3 A5	4 A4	5 A3	6 A2	7 A1	8 PR	Switch Position Address Line
00	0	0	0	0	0	0	0	0	0 = Closed (ON)
01	0	0	0	0	0	0	0	1	1 = Open (OFF)
02	0	0	0	0	0	0	1	0	
03	0	0	0	0	0	0	1	1	
04	0	0	0	0	0	1	0	0	
05	0	0	0	0	0	1	0	1	
06	0	0	0	0	0	1	1	0	
07	0	0	0	0	0	1	1	1	
08	0	0	0	0	1	0	0	0	
09	0	0	0	0	1	0	0	1	
0A	0	0	0	0	1	0	1	0	
0B	0	0	0	0	1	0	1	1	
0C	0	0	0	0	1	1	0	0	
0D	0	0	0	0	1	1	0	1	
0E	0	0	0	0	1	1	1	0	
0F	0	0	0	0	1	1	1	1	
10	0	0	0	1	0	0	0	0	
.									
.									
.									
.									
.									
E0	1	1	1	0	0	0	0	0	
E1	1	1	1	0	0	0	0	1	
.									
FF	1	1	1	1	1	1	1	1	

NOTE: Software from SSM will use ports E0 and E1 for the keyboard status and data ports, respectively.

PR = Port Reverse Switch

4.4 ADDRESS STROBE OPTION

The address decoder IC for the on-board RAM is an 8131 (U25). This part can be strobed to latch the address in addition to a simple compare function.

Standard 8080 or Z80 CPU	
with 8080 timing:	connect E6 to E7 (strobed)
Z80 timing:	connect E5 to E6 (not strobed)
IEEE 696 timing:	see special NOTE in this section

This strobe is generated by NANDing the Ø1 clock (S-100 bus pin 25) and PSYNC (S-100 bus pin 76) and then applying this signal to the enable input of U25. Both of these signals are true (Logic 1) during a valid memory address, thereby, generating a logic 0 enable signal from the output of U30 pin 3. Certain Z-80 CPU boards could use the strobed addressing option (8080 timing), the SSM CB-2, for example, could be used in this mode. To determine if your CPU can be used with the strobe option, check your CPU board manual to be sure the timing relationships between Ø1, PSYNC, and the address bus are correct.

The latching action will reduce the possibility of bus noise creating a valid address and blanking the screen when not wanted. It is recommended, in all cases, to have a terminated S-100 motherboard or use a terminator board (such as the SSM T1) to reduce noise and prevent floating signal lines in the computer.

NOTE: The new IEEE standard has created a new signal on that bus which replaces Ø1 (pin 25). This new signal is called $\overline{\text{PSTVAL}}$ and is similar to PSYNC nanded with Ø1. If your CPU generates PSTVAL, tie pin E6 on the VB3 to U30, pin 1 with a jumper wire.

4.5 KEYBOARD CONNECTOR

The keyboard is connected to the VB3 board using a 14 pin DIP socket. The pin assignments are as follows:

PIN 1	Positive strobe input from keyboard (see note)
PIN 2	$\overline{\text{INT}}$
PIN 3	Data Bit 7
PIN 4	Data Bit 5
PIN 5	Data Bit 3
PIN 6	Data Bit 1
PIN 7	Signal Ground
PIN 8	Signal Ground
PIN 9	Data Bit 0
PIN 10	Data Bit 2
PIN 11	Data Bit 4
PIN 12	Data Bit 6
PIN 13	Not Used
PIN 14	Not Used

NOTE: The strobe pulse generated by the keyboard should have a minimum duration of 100 nanoseconds.

4.6 DISPLAY MEMORY ADDRESS

The VB3 uses up to 8K of address space to hold information for its video display and attributes. This on-board RAM can be addressed to any 8K boundary and is determined by the setting of switch S2 positions 1, 2, and 3 as follows:

	S2 SWITCH POSITION		
LOCATION	1	2	3
0000-1FFF	0	0	0
2000-3FFF	0	0	1
4000-5FFF	0	1	0
6000-7FFF	0	1	1
8000-9FFF	1	0	0
A000-BFFF	1	0	1
C000-DFFF	1	1	0
E000-FFFF	1	1	1

0 = Closed (ON)
1 = Open (OFF)

NOTE: Software obtained from SSM will use C000 through DFFF for the video display memory.

4.7 CRT CONTROLLER I/O PORTS

The VB3 uses a software programmable CRT controller chip to control the video display timing and the display format. Programming is accomplished through 16 I/O ports. This address can be set to any 16 port boundary and is determined by the setting of switch S4 positions 4, 5, 6, and 7 as follows:

	S4 SWITCH POSITION			
LOCATION	4	5	6	7
00-0F	0	0	0	0
10-1F	0	0	0	1
20-2F	0	0	1	0
30-3F	0	0	1	1
40-4F	0	1	0	0
50-5F	0	1	0	1
60-6F	0	1	1	0
70-7F	0	1	1	1
80-8F	1	0	0	0
90-9F	1	0	0	1
A0-AF	1	0	1	0
B0-BF	1	0	1	1
C0-CF	1	1	0	0
D0-DF	1	1	0	1
E0-EF	1	1	1	0
F0-FF	1	1	1	1

0 = Closed (ON)
1 = Open (OFF)

NOTE: Software obtained from SSM will use ports D0-DF for addressing the CRT controller.

4.8 EPROM SET-UP (2716 OR 2732)

Wire-wrap pads allow the EPROM located at U28 to be either a 2716 (2K x 8) or a 2732 (4K x 8). The 2716 will allow 128 user defined characters, while the 2732 will allow 256 characters. The EPROM type is selected as follows:

2716: Connect E2 and E4
2732: Connect E3 and E4

4.9 VIDEO OUTPUT CONNECTOR

The output connector is located in the upper left corner of the PC board. It has signals for composite video, parallel video, and sync input.

Pin 3: Composite video output
Impedance = 50 to 75 ohms
Level = approximately 1.25v peak-to-peak (with 75 ohms)

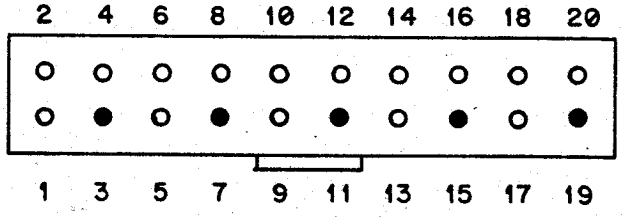
Pin 7: Vertical drive output
Impedance = 1K ohms
Level = approximately 4.5v peak-to-peak

Pin 11: Horizontal drive output
Impedance = 1K ohms
Level = approximately 4.5v peak-to-peak

Pin 15: Horizontal sync input
Impedance = 10K ohms
Level = +0.4v to +2.4v (+5v maximum)

Pin 19: Vertical sync input
Impedance = 10K ohms
Level = +0.4v to +2.4v (+5v maximum)

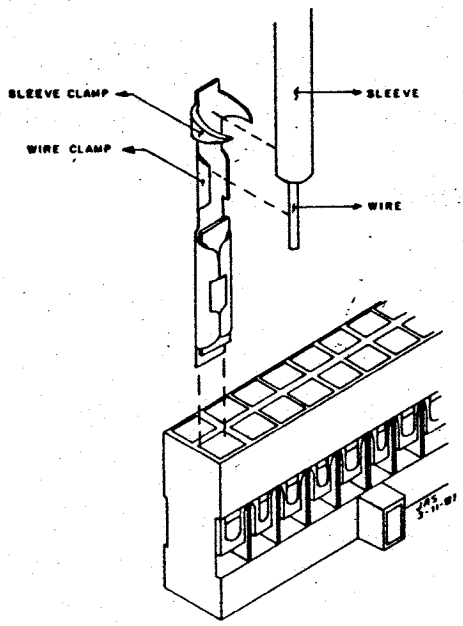
All other pins are grounded



Connector pattern (top view)

The connector shell is provided with ten crimp-type pins to allow signal and ground for the five main signals. The connector shell is keyed so that it cannot be placed on the header with the pins reversed.

The pins should be installed as follows:



4.10 BANK SELECT

The VB3 is set up for I/O bank selecting with the keyboard status and data ports on the VB3 used as a control port. Bank select gives the user the ability of (1) using multiple VB3's at the same memory location with the same address for the CRT I/O control ports, and (2) running a 64K CP/M system where the VB3 becomes invisible to the S-100 bus when not in use, allowing full use of all 64K for user programs.

To activate the bank select feature, install IC U19 (74LS74). When U19 is installed, the **keyboard status port address** will have to be **written to** whenever you want to access the VB3 memory or control ports. You may output any data value to this port.

To deactivate the VB3 board, the **keyboard data port address** will have to be **written to.** Again, you may output any data value to this port.

For multiple VB3's, occupying the same memory space, the keyboard port address for each VB3 board must be different. The VB3's can all use the same software driver routine (one routine), but before calling it you must activate the desired VB3, and then deactivate it when the routine returns.

For a 64K RAM based computer system, the memory that the VB3 will overlay **MUST** be equipped with a Phantom Disable feature (Bus pin 67). Also, the VB3 software driver and the scratch RAM used for stack operations **cannot** overlay the VB3, since it would disable its own program during video memory accesses.

If you are using the bank select feature in an interrupt driven system, be sure to disable the interrupts before using the VB3 output routine, and then enable the interrupts after turning the VB3 off with bank select. This prevents problems if the VB3 just happens to be overlaying your interrupt handler routine.

4.11 MWRITE OPTION

During the assembly procedure, the manual instructs you not to install R19 and C16. If you desire the capability to use the MWRITE signal (Bus pin 68) to deposit into the on-board RAM (instead of $\overline{PWR}$), then install R19. If noise is present on this signal, then install C16 to provide filtering. (It is recommended in the IEEE 696 standard that MWRITE not be used as a memory write control signal. However, some older S-100 systems may require this (especially those with front control panels).

4.12 STANDARD HARDWARE CONFIGURATION

The software supplied with this document assumes that the hardware has been configured in the following configuration:

SWITCH	POSITION	SETTING	DESCRIPTION
S1	1	Open (1)	External Sync is not used
S1	2	Open (1)	
S2	4	Open (1)	9 dots per character
S2	3	Open (1)	
S2	2	Open (1)	
S2	1	Closed (0)	
S3	1	Open (1)	KDATA is at E1 (hex)
S3	2	Open (1)	KSTAT is at E0 (hex)
S3	3	Open (1)	
S3	4	Closed (0)	
S3	5	Closed (0)	
S3	6	Closed (0)	
S3	7	Closed (0)	
S3	8	Open (1)	
S4	1	Open (1)	Display memory starts at C000
S4	2	Open (1)	
S4	3	Closed (0)	
S4	4	Open (1)	CRT controller starts at D0
S4	5	Open (1)	
S4	6	Closed (0)	
S4	7	Open (1)	

JUMPERS

E2 to E4 for 2716 EPROM in position U28

E5 to E6 or E6 to E7 depending on 8080 or Z-80 CPU timing

5.0 HARDWARE INFORMATION

Refer to the SMC 5037 data sheet in the APPENDIX for additional information. In the following sections 'N' refers to the base I/O port address set by switch S4 positions 4, 5, 6, and 7 (Section 4.7).

5.1 CRT CONTROLLER REGISTERS

The VB3 uses a CRT controller chip to control the video timing and display format. The chip contains 8 hardware registers that are programmed through the 16 I/O ports addressed by Switch 4 (Section 4.7). Registers 0 through 6 are write-only registers and are loaded through I/O ports N+0 through N+6. Registers 7 and 8 can be loaded through ports N+12 and N+13 or read through ports N+9 and N+8 respectively. I/O ports N+10 and N+14 will reset or start the CRT controller. I/O port N+11 will cause register 6 to be incremented, scrolling the screen. I/O ports N+7 and N+15 are not presently used in the VB3 design.

Hardware Register 0 (I/O port N+0: Write Only)

This register contains the number of characrer times for 1 horizontal period of the TV raster scan.

The number of character times per horizontal scan (HCOUNT) is given by:

HCOUNT= DCLK/DOTC/SCANF/FRAMS

where DCLK is the dot clock in units of dots per second
DOTC is the size of a character in dots per character
SCANF is the number of scans per frame
FRAMS is the frames per second in frames per second

The number that should be programmed into Register 0 is HCOUNT-1.

Some restricions are imposed on the parameters. These restrictions are the result of hardware limitations.

The dot clock is limited by the quality of the monitor. If the dot clock is too fast, then the characters will tend to blur together.

The number of dots per character should be around 8 to 10 for a nice looking display. A character size of 9 seems to give alphanumeric data their best appearance, while a size of 10 will work best for the wide graphics mode. A character size of 8 can be used if you are trying to squeeze more data onto the screen.

The number of scans-per-frame must be at least 513 for interlaced and 256 for non-interlaced modes. At most, they should not exceed about 540 for interlaced and 270 for non-interlaced unless you are using a high quality monitor or are on the European TV rates. In addition, the number of scans-per-frame must be odd for interlaced and even for non-interlaced modes.

The number of frames-per-second must be 30 (25 European) for interlaced and 60 (50 European) for non-interlaced modes. Other values will result in a waving effect on the screen.

Example: Assume the following values:
DCLK = 16,000,000 dots/second
DOTC = 9 dot/character
SCANF= 525 scan/frame
FRAMS= 30 frames/second

then:

$$\text{HCOUNT} = \frac{16{,}000{,}000 \text{ dots}}{\text{second}} * \frac{\text{char}}{9 \text{ dots}} * \frac{\text{frame}}{525 \text{ scans}} * \frac{\text{second}}{30 \text{ frame}}$$

= 113 char/scan (rounded to the nearest integer)

The value programmed into Register 0 is HCOUNT-1 or 112.

Hardware Register 1 (I/O port N+1: Write Only)

This register contains 3 fields of information. The most significant bit (D7) is the interlace bit. D7 = 1 means interlaced and D7 = 0 means non-interlaced mode.

The next 4 bits (D6, D5, D4, and D3) determine the size of the horizontal sync pulse (HSP). A typical value of 8 will do nicely.

The last 3 bits (D2, D1, and D0) determine the time between the horizontal pulse (HBP) and the beginning of data. The value here is not critical and can be used to position the data horizontally on the screen.

NOTE: HCOUNT > HSP + HBP + number of characters per line

Hardware Register 2 (I/O port N+2: Write Only)

This register defines the number of scans per character (typically 13 for the SMC 8002 character set and 10 for the alternate character set contained in U28) and the number of characters per row (typically 80). Bit 6 through 3 should contain the value SCANR-1 (where SCANR is the number of scans per row).

Bits 2 through 0 contain a 3 bit code for the number of characes per line. Using the standard 80-character mapper PROMs (U2 and U3), the value 5 should be programmed into this field. The code for other screen sizes are:

SIZE	CODE
20	0
32	1
40	2
64	3
72	4
80	5
96	6
132	7

NOTE: Mapper PROMs U3 and U4 are set for 80 columns in the standard VB3.

Hardware Register 3 (I/O port N+3: Write Only)

This register contains the skew bits and the number of data-rows-per-frame. The VB3 board has been designed to require no skewing adjustments, therefore, the skew bits (D7 and D6) should be 0.

The 6 least significant bits (D5 through D0) define the number of data-rows-per-frame. The register should contain NROWS-1, where NROWS is the number of data-rows-per-frame.

Hardware Register 4 (I/O port N+4: Write Only)

This register defines the number of scans-per-frame. In interlaced mode, the register should contain:

$$(SCANF - 513)/2$$

In interlaced mode, the register should contain:

$$(SCANF - 256)/2$$

Hardware Register 5 (I/O port N+5: Write Only)

This register contains the number of scan lines between the beginning of the vertical sync pulse and the start of the first data row (top margin). Increasing this value will position the data lower on the screen. The following formula has been used successfully to center the data on the display:

$$\text{Vertical Scan Delay} = (SCANF - NROWS * SCANR)/4 + C$$

where SCANF = the number of scans per frame
NROWS = the number of data rows
SCANR = the number of scans per data row
C = some constant that seems to work for your monitor

The value for C is approximately 8.

Hardware Register 6 (I/O port N+6: Write Only)

This register contains the row address of the last row displayed on the screen. For example, if the display was set for 24 rows and this register contained a 15, then row 15 would be at the bottom of the screen and row 16 would be at the top of the screen. Rows 0 and 23 would be in the middle of the screen.

Hardware Register 7 (I/O port N+9: Write Only; N+12: Read Only)

This register contains the column address of the hardware cursor. It may be written through I/O port 12 or read through I/O port 9. A value of 0 will put the cursor on column 1.

Hardware Register 8 (I/O port N+8: Write Only; N+13: Read Only)

This register contains the row address of the hardware cursor. It may be written through I/O port 13 or read through I/O port 8. A value of 0 will put the cursor on row 1. The actual location of the cursor as seen on the screen will also depend on the contents of register 6, which determines where the selected row will physically be displayed.

5.2 USER DEFINED CHARACTER SET

The VB3 can be programmed, using the attribute byte, to use a 2716/2732 EPROM (not supplied) as the character generator. Up to 256 user-defined characters are available using a 2732 EPROM (128 using a 2716). With the EPROMs, 16 bytes are used for each character. Each byte represents one scan of the character, with up to 16 scans per character. For each byte, bit 7 is shifted out first and will appear in the left-most part of the character. For example, to program the letter "E" to by display for the code of 045H, program the following at the indicated locations of the EPROM:

LOCATION (IN HEX)	OUTPUT BYTE 7	6	5	4	3	2	1	0
450	0	0	0	0	0	0	0	0
451	0	1	1	1	1	1	1	1
452	0	1	0	0	0	0	0	0
453	0	1	0	0	0	0	0	0
454	0	1	0	0	0	0	0	0
455	0	1	1	1	1	0	0	0
456	0	1	0	0	0	0	0	0
457	0	1	0	0	0	0	0	0
458	0	1	0	0	0	0	0	0
459	0	1	1	1	1	1	1	1
45A	0	0	0	0	0	0	0	0
45B	0	0	0	0	0	0	0	0
45C	0	0	0	0	0	0	0	0
45D	0	0	0	0	0	0	0	0
45E	0	0	0	0	0	0	0	0
45F	0	0	0	0	0	0	0	0

NOTE: In most cases, the VB3 is set to display an 8x13 or 9x13 matrix, so only 13 bytes out of 16 possible will be used for each character.

5.3 NEW LINE LENGTHS

If you have a requirement for a screen size other than 80 characters-per-line (such as 64 or 132), contact SSM for the required optional mapper PROMs. These new optional mapper PROMs will replace U3 and U4.

5.4 INTERLACED VS. NON-INTERLACED VIDEO

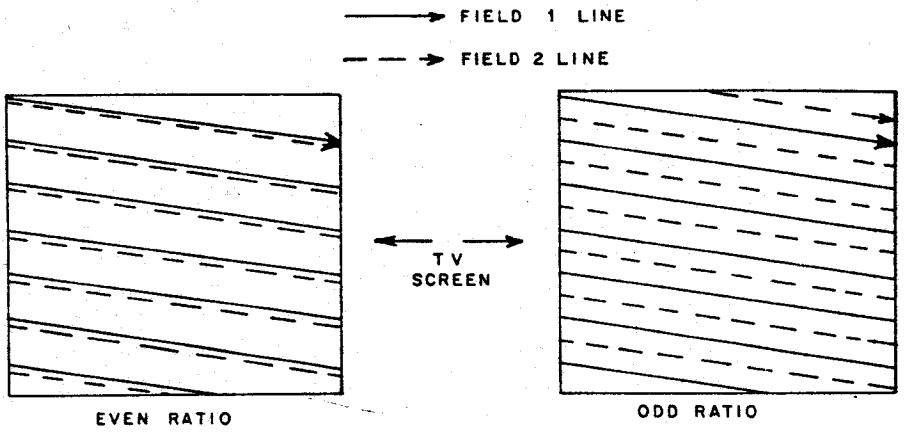
First, what is interlace?

The video picture is created by hundreds of image lines drawn horizontally on the phosphor-coated face of the picture tube by an electron beam. If the intensity of the electron beam is varied during the image line for several lines, a shape (image) is created on the picture tube.

The ratio of vertical to horizontal timing rates sent in a composite video signal will determine the number of lines used vertically to display a picture.

(a) If the ratio is even, then every image (field) will be drawn over the top of the lines of the previous image. At U.S. TV rates, a new field will be started every 1/60 second and a complete frame (2 fields) will be generated every 1/30 second.

(b) If the ratio is odd, then a series of lines plus one fractional line will be drawn in one field. The second field will continue the fractional line along with the full length lines to fill out itself. If the fraction is 1/2 a line, then the two fields will interleave (interlace) every other line in the video display. To the observer, there are now twice as many lines in the video display during every complete frame than in the even ratio (a) case.



An odd number of lines is used in a standard TV sync signal, so every other line of an image is drawn in one field, and while it is fading out the second half (every other line) is drawn in the second field. Since the human eye will store an image for a short period of time, the two fields get integrated into one relatively smooth sequence of scenes for large images. Interlace is a way to double the number of visible lines (vertical resolution) in a video display.

What is the advantage of interlace

Interlace will double the number of character lines available in the video display for word processing or graphic pictures. Let look at an example: 12 scans per character

Case 1 TV Rates: Horizontal = 15720 Hz
Vertical = ÷ 60 Hz
No-Interlace 262 lines (ratio) (EVEN)

Character lines possible = $\frac{262}{12}$ = 21.88 lines maximum

But, vertical blanking uses some lines, so you will be lucky to get above 20 character lines maximum.

Case 2 TV Rates: Horizontal = 15750 Hz
Vertical = ÷ 60 Hz
Interlaced 262.5 lines (ratio) (ODD)

Character lines possible = $\frac{262.5}{12}$ x 2 = 43.75 lines maximum

But, vertical blanking uses some lines, so you should be able to get about 40 character lines or less.

What is the disadvantage to interlace?

Using interlace relies on the eye and the TV screen to hold an image long enough so that the mind doesn't realize that the image is being re-created every 1/30 second (one frame). For a large image (many lines), this works reasonably well. For words only 12 lines high (one character), with the eye focusing on a much smaller piece of the screen, the mind becomes more aware that 6 lines are changing between 6 lines (interlace). This interlace process has been called by some hobbyists "jitter" or "twinkle". To reduce or eliminate this phenomenon, the picture tube must be made to hold a field longer (persistence) while the next is being drawn. Manufacturers of quality TV monitors make available a wide variety of phosphors for a picture tube, when requested. Normal TV tubes use a "P4" phosphor for viewing, which is too short a persistence for interlaced characters. It is recommended that a longer persistence phosphor be used for interlace.

PHOSPHOR	COLOR	BRIGHTNESS	PERSISTENCE
P40	White	medium	medium
P42	Green	medium to high	medium
P39	Green	low	long

If you don't want interlace, but you do want additional character lines in the text display, refer to section 5.6.

5.5 COMPOSITE VIDEO VS. SEPARATE VIDEO

The TV monitors available on the market divide into two basic types:

Type I : Monitor with a composite video input
Type II: Monitor with seperate video & sync inputs

The Type I monitor uses a single incoming signal to present video image information and TV timing to its circuitry. The incoming signal (composite) has the horizontal and vertical timing, as pulses, mixed with the video image. The level of this signal is normally 1 volt (peak to peak) when terminated into a 75 ohm load. The Type I monitor assumes that the horizontal rate is approximately 15750 Hz (15625 Hz European) and the vertical rate is approximately 60 Hz (50 Hz European), so that the circuitry can detect these signals needed for correct synchronization. The Type I monitor will normally not accept a significant deviation from the standard TV rates without losing synchronization. The following scan lines are possible with a Type I monitor:

Standard	Horizontal Rate	Vertical Rate	Non-Interlaced Lines	Interlaced Lines
U.S.	15750	60	~ 262	525
European	15625	50	~ 312	625

The Type II monitor uses two or three incoming signal lines to receive the required information. The video information is sent to the monitor seperately from the TV timing signals. Therefore, the Type II monitor does not have to detect and seperate the timing signals as does the Type I monitor, so it will usually accept a wider variation of the horizontal and vertical rates. If the vertical rate is held to the standard rate, but the horizontal rate is increased to more than the standard rate, than more scan lines are added to the screen. More scan lines produce more text lines, at the ratio of one text line for every group of 12 additional scan lines from the VB3. If the VB3 is used for only upper case characters, then only 10 scan lines are needed per character. This increases the number of text lines by 16%. (Note: 10 scan lines will not work well for graphics, which needs a number divisible by 4).

In summary, a Type II monitor can be driven to higher rates (per the manufacturer) for more character lines, compared to more Type I monitors, giving the user more choices in using non-interlace or interlace displays.

5.6 CHARACTER MATRIX COMBINATIONS

The VB3 CRT controller port address is represented by two hex digits, where the most significant hex digit is set by switch S4 (see Section 4.7), and the least significant hex digit (P, used in the table below) describes a specific address where data is sent. To set up the described character matrix (like 80 x 16), different data bytes (70, 65, 5D, etc.) must be sent to each specific port address (P0, P1, P2, etc.) upon initialization of the VB3.

For example, to set up a system for 80 x 24, U.S. standard, full interlace, when the beginning port address is D0, you will have to send 70 Hex to port D0, BC Hex to port D1, 6D to port D2, 17 Hex to port D3, 06 Hex to port D4, 29 Hex to port D5, 17 Hex to port D6, 00 Hex to port DC, and 17 Hex to port DD.

Examples:
U.S. = Horizontal (15750 Hz) Vertical (60 Hz)
European= Horizontal (15625 Hz) Vertical (50 Hz)
NA = Not applicable

CHARACTER MATRIX	TV RATE	INTERLACED	'P' DIGITS 0	1	2	3	4	5	6	C	D	NOTES
80x16	U.S.	no	70	65	5D	0F	03	26	0F	00	0F	1,6,7
80x18	U.S.	no	70	65	5D	11	03	1C	11	00	11	1,6,7
80x20	European	no	71	65	5D	13	1C	25	13	00	13	1,6,7
80x23	European	no	71	65	5D	16	1C	14	16	00	16	1,6,7
80x24	U.S.	no	70	69	4D	17	03	0C	17	00	17	1,2,6,8
80x24	U.S.	yes	70	BC	6D	17	06	29	17	00	17	1,6,8
80x32	U.S.	yes	70	D3	5D	1F	06	1E	1F	00	1F	1,5,8
80x33	U.S.	yes	70	BC	6D	20	06	0F	20	00	20	1,5,9
80x36	U.S.	yes	70	D3	5D	23	06	16	23	00	23	1,5,9
80x40	European	yes	71	D3	5D	27	38	22	17	00	27	1,5,9
80x46	European	yes	71	D3	5D	2A	38	12	2A	00	2A	1,5,9
80x24	NA	no	5F	0C	5D	17	1A	0D	17	00	17	1,8,10
80x38	U.S.	yes	70	BC	5D	25	06	09	25	00	25	1,9,10
80x38	NA	yes	69	9B	5D	25	17	10	25	00	25	1,9,10
80x48	NA	yes	69	9B	5D	2F	4F	0E	2F	00	2F	1,9,10
80x50	NA	yes	70	E9	45	31	06	0C	31	00	31	1,2,9,10
64x16	NA	no	6D	3C	6B	0F	07	0F	0F	00	0F	1,3,6,7
132x28	NA	yes	A1	BC	6F	19	00	1F	19	00	19	1,3,4,5,9

NOTES:
(1) Character size (width) is set for 9 dots/character. (Except where noted)
(2) Requires use of alternate character EPROM (6 x 7 matrix) using a '01' attribute byte.
(3) Need new mapper PROM's for U3 and U4. Contact factory.
(4) Crystal Y1 changed to 20 MHz for more band width. Character size set to 8 dots/character.
(5) Requires P39 phosphor monitor.
(6) P4 phosphor monitor.
(7) 9" or larger monitor recommended.
(8) 12" or larger monitor recommended.
(9) 15" or larger monitor recommended.
(10) Requires driven monitor. (Seperate video and sync inputs)

5.7 ATTRIBUTE BYTE

For every character in the VB3 memory that is displayed, there is an attribute byte that controls how it will be displayed. The 8K is divided into two sections: (1) the first 4K of memory is for characters, and (2) the second 4K of memory is for attributes. The VB3, when reading from memory re-maps the memory into 16-bit words that combine each character and its associated attribute bytes.

Each attribute byte controls about ten different ways its associated character can be displayed. The effect of each bit is as follows:

			D1	D0		
Bit	D0	Coded	0	0	=	Graphics (2x4 per character)
Bit	D1	Coded	0	1	=	Alternate ROM (2716/2732)
			1	0	=	Thin-line graphics
			1	1	=	Alpha (128 ASCII characters)

Bit D2 Invert Video
Bit D3 Blank-Out character
Bit D4 Underline character
Bit D5 Flash character
Bit D6 Strike-thru character
Bit D7 Gray level

The bits in the attribute byte can be mixed to any combination to meet the user's needs. Each attribute byte is 1000 hex (4K) above the character it will affect.

Refer to the SMC 8002 Video Display-Controller Video Generator data sheets in the APPENDIX for a description of graphic and thin-line graphics.

6.0 SOFTWARE

6.1 CP/M Compatible Driver Routine

```
                ;"VBIOS"; Rev 0; Written by Ben L Gee; Dec 9, 1979
                ;Modified by Dan Fischler on August 21, 1980

                ;VBIOS is a driver for the SSM VB3 video board.  VBIOS contains
                ;only enough code to run CP/M and some CP/M programs. If a more
                ;intelligent driver is required then you should consider ICRT.
                ;

                ;       DEFINE SOME SYSTEM EQUATES.
E000 =          ROM     EQU     0E000H          ;Beginning of ROMmable code
0000 =          FALSE   EQU     0
FFFF =          TRUE    EQU     NOT FALSE

                ;       DEFINE THE VB3 PARAMETERS
C000 =          VIDEO   EQU     0C000H  ;Address of video memory
1000 =          OFFSET  EQU     01000H  ;Relative offset to attribute field
00D0 =          VTAC    EQU     0D0H    ;I/O address of the CRT controller
00E0 =          KSTAT   EQU     0E0H    ;Keyboard Status/Board Enable Port
00E1 =          KDATA   EQU     0E1H    ;Keyboard Data/Board Disable Port
0050 =          NCOLS   EQU     80      ;Number of Columns
0018 =          NROWS   EQU     24      ;Number of Rows
0000 =          SKEW    EQU     0
020D =          SCANF   EQU     525     ;Scans per frame
000B =          SCANR   EQU     11      ;Scans per data row
0071 =          HCOUNT  EQU     113     ;Character times in 1 horiz scan line
FFFF =          INTERL  EQU     TRUE    ;Display is interlaced
0005 =          CODE    EQU     5       ;80 columns
0003 =          NORMAL  EQU     3       ;Set alphanumeric mode



                ;       DEFINE SOME ASCII VALUES
0008 =          BS      EQU     08H
000A =          LF      EQU     0AH
000D =          CR      EQU     0DH


                ;       VIDEO DRIVER SUBROUTINES
                ;
                ;       ENTRIES POINTS ARE:
                ;
                ;               VBINIT - To init the hardware and software
                ;               VBIN - To get a keystroke from the keyboard
                ;               VBOUT - To output a character to the display
                ;               VBSTAT - To check the keyboard status
                ;
                ;       CALLING SEQUENCES:
                ;
                ;               CALL    VBINIT
                ;
                ;               CALL    VBIN
                ;               STA     KEYSTROKE       ;Result in Register A
                ;
```

```
                ;               LDA     DATA            ;Output char in Reg. C
                ;               MOV     C,A
                ;               CALL    VBOUT
                ;
                ;               CALL    VBSTAT          ;Returns Reg. A=0 if
                ;                                       ;No key is available,
                ;                                       ;else Returns Register
                ;                                       ;A=0FFH
                ;
E000                    ORG     ROM
E000 C30CE0             JMP     VBINIT
E003 C345E0             JMP     VBIN
E006 C34EE0             JMP     VBOUT
E009 C33CE0             JMP     VBSTAT


E00C D3E0       VBINIT: OUT     KSTAT           ;Enable the VB3 board
E00E 210000             LXI     H,0
E011 CDDDE0             CALL    EEOS            ;Clear the screen
E014 CDE8E0             CALL    CURON           ;Put the cursor on the screen

                ; INITALIZE ALL OF THE CONTROL REGISTERS

E017 D3DE               OUT     VTAC+14
E019 D3DA               OUT     VTAC+10         ;Reset the VTAC

E01B 3E70               MVI     A,HCOUNT-1
E01D D3D0               OUT     VTAC+0

                        IF      NOT INTERL
                        MVI     A,03CH
                        ENDIF
                        IF      INTERL
E01F 3EBC               MVI     A,0BCH
                        ENDIF
E021 D3D1               OUT     VTAC+1

                        IF      NOT INTERL
                        MVI     A,(SCANR-1)*8+CODE
                        ENDIF
                        IF      INTERL
E023 3E4D               MVI     A,(SCANR-2)*8+CODE
                        ENDIF
E025 D3D2               OUT     VTAC+2

E027 3E17               MVI     A,SKEW*64+NROWS-1
E029 D3D3               OUT     VTAC+3

                        IF      INTERL
E02B 3E06               MVI     A,(SCANF-513)/2
                        ENDIF
                        IF      NOT INTERL
                        MVI     A,(SCANF-256)/2
                        ENDIF
```

```
:02D D3D4               OUT     VTAC+4

:02F 3E49               MVI     A,(SCANF-NROWS*SCANR)/4+8
:031 D3D5               OUT     VTAC+5          ;Try to center the data on
                                                ;the screen


:033 3E17               MVI     A,NROWS-1
:035 D3D6               OUT     VTAC+6

:037 D3DE               OUT     VTAC+14         ;Start the timing chain
:039 D3E1               OUT     KDATA           ;Disable the VB3 board
:03B C9                 RET


:03C DBE0       VBSTAT: IN      KSTAT           ;Check keyboard status
:03E 2F                 CMA
:03F E601               ANI     1
:041 C8                 RZ                      ;Return a zero if no data is
:042 3EFF               MVI     A,0FFH          ;available
:044 C9                 RET                     ;Else return a 0FFH

:045 CD3CE0     VBIN:   CALL    VBSTAT          ;Return the next key from the
                                                ;keyboard
:048 CA45E0             JZ      VBIN            ;Wait until one is available
:04B DBE1               IN      KDATA           ;Get it
:04D C9                 RET                     ;And return


                ; VBOUT IS THE MAIN DISPLAY DRIVER FOR THE VB3 BOARD
:04E E5         VBOUT:  PUSH    H               ;First, save all registers
:04F D5                 PUSH    D
:050 C5                 PUSH    B
:051 D3E0               OUT     KSTAT           ;And enable the VB3 board
:053 CDF1E0             CALL    CUROFF          ;Then, turn off the cursor
:056 CD62E0             CALL    PROCES          ;And process the character
:059 CDE8E0             CALL    CURON           ;Now, turn the cursor back on,
:05C D3E1               OUT     KDATA           ;disable the VB3 board,
:05E C1                 POP     B               ;and restore the registers
:05F D1                 POP     D
:060 E1                 POP     H
:061 C9                 RET

                ; ON ENTRY, REG HL = LOGICAL CURSOR ADDRESS AND REG C = DATA.
:062 79         PROCES: MOV     A,C             ;Move the data to Register A
:063 E67F               ANI     7FH             ;Strip parity and set flags
:065 C8                 RZ                      ;Skip nulls
:066 FE0D               CPI     CR              ;Check for control characters
:068 CA83E0             JZ      DOCR            ;If match, then do control
:06B FE0A               CPI     LF              ;character
:06D CA86E0             JZ      DOLF
:070 FE08               CPI     BS
:072 CABBE0             JZ      DOBS
```

```
                ; MUST BE A DATA CHARACTER.
E075 E5                 PUSH    H               ;Save the logical cursor addr.
E076 CDFCE0             CALL    GETBA           ;Get the physical cursor addr.
E079 71                 MOV     M,C             ;Put the data there
E07A 110010             LXI     D,OFFSET
E07D 19                 DAD     D               ;Compute the address of the
                                                ;governing attribute
E07E E1                 POP     H               ;Get the logical cursor
                                                ;address back
E07F CDD5E0             CALL    RIGHT           ;Move the cursor right
E082 C0                 RNZ                     ;Done if the cursor didn't
                                                ;wrap around

                ; PUT THE CURSOR ON COLUMN 1 OF THE CURRENT LINE.
E083 2E00       DOCR:   MVI     L,0             ;Put the cursor on Col 1 of
                                                ;the next line
E085 C9                 RET

                ; PUT THE CURSOR ON THE NEXT LINE, BLANK THE LINE.
E086 CDC6E0     DOLF:   CALL    DOWN            ;Move the cursor down one line
E089 E5                 PUSH    H               ;Save the cursor address
E08A 2E00               MVI     L,0             ;Put the cursor on column 1
E08C CD91E0             CALL    EEOL            ;Erase to the end of the line
E08F E1                 POP     H               ;Restore the cursor address
E090 C9                 RET                     ;Return to caller

                ; ERASE FROM THE CURSOR TO THE END OF THE CURRENT LINE.
E091 3E50       EEOL:   MVI     A,NCOLS
E093 95                 SUB     L
E094 47                 MOV     B,A
E095 0E01               MVI     C,1

                ; ERASE THE SCREEN BEGINNING AT THE CURSOR POSITION.  THE
                ; NUMBER OF CHARACTERS TO ERASE IS IN REG BC.
E097 E5         ERASE:  PUSH    H
E098 E5         ERASE1: PUSH    H
E099 CDFCE0             CALL    GETBA
E09C 110010             LXI     D,OFFSET
E09F EB                 XCHG
E0A0 19                 DAD     D
E0A1 EB                 XCHG
E0A2 3E03               MVI     A,NORMAL        ;Use the normal attribute
E0A4 3620       ERASE2: MVI     M,' '
E0A6 12                 STAX    D
E0A7 23                 INX     H
E0A8 13                 INX     D
E0A9 05                 DCR     B
E0AA C2A4E0             JNZ     ERASE2
E0AD E1                 POP     H
E0AE 2E00               MVI     L,0
E0B0 CDC6E0             CALL    DOWN
E0B3 0650               MVI     B,NCOLS
E0B5 0D                 DCR     C
E0B6 C298E0             JNZ     ERASE1
E0B9 E1                 POP     H
```

```
E0BA C9                 RET

                ; BACK THE CURSOR BY ONE POSITION.
E0BB CDCEE0     DOBS:   CALL    LEFT
E0BE C0                 RNZ

                ; MOVE THE CURSOR UP.
E0BF 7C         UP:     MOV     A,H
E0C0 25                 DCR     H
E0C1 B7                 ORA     A
E0C2 C0                 RNZ
E0C3 2617               MVI     H,NROWS-1
E0C5 C9                 RET

                ; MOVE THE CURSOR DOWN.
E0C6 24         DOWN:   INR     H
E0C7 3E18               MVI     A,NROWS
E0C9 BC                 CMP     H
E0CA C0                 RNZ
E0CB 2600               MVI     H,0
E0CD C9                 RET

                ; MOVE THE CURSOR LEFT.
E0CE 7D         LEFT:   MOV     A,L
E0CF 2D                 DCR     L
E0D0 B7                 ORA     A
E0D1 C0                 RNZ
E0D2 2E4F               MVI     L,NCOLS-1
E0D4 C9                 RET

                ; MOVE THE CURSOR RIGHT.
E0D5 2C         RIGHT:  INR     L
E0D6 3E50               MVI     A,NCOLS
E0D8 BD                 CMP     L
E0D9 C0                 RNZ
E0DA 2E00               MVI     L,0
E0DC C9                 RET


                ; ERASE FROM THE CURSOR POSITION TO THE END OF THE SCREEN.
E0DD 3E50       EEOS:   MVI     A,NCOLS
E0DF 95                 SUB     L
E0E0 47                 MOV     B,A
E0E1 3E18               MVI     A,NROWS
E0E3 94                 SUB     H
E0E4 4F                 MOV     C,A
E0E5 C397E0             JMP     ERASE

                ; TURN THE CURSOR ON.
                ; THE LOGICAL ADDRESS OF THE CURSOR MUST BE IN REG HL.
E0E8 7D         CURON:  MOV     A,L
E0E9 D3DC               OUT     VTAC+12
E0EB 7C                 MOV     A,H
E0EC D3DD               OUT     VTAC+13
```

```
E0EE D3D6               OUT     VTAC+6          ;Make sure the cursor is on
E0F0 C9                 RET                     ;the bottom

                ; TURN THE CURSOR OFF.
                ; THE LOGICAL ADDRESS OF THE CURSOR IS RETURNED IN REG HL.
E0F1 DBD9       CUROFF: IN      VTAC+9          ;Read the column register
E0F3 6F                 MOV     L,A             ;To Register L
E0F4 DBD8               IN      VTAC+8          ;Read the row register
E0F6 67                 MOV     H,A             ;To Register H
E0F7 3EFF               MVI     A,0FFH          ;Move the cursor off the screen
E0F9 D3DC               OUT     VTAC+12
E0FB C9                 RET

                ; CONVERT A LOGICAL ADDRESS TO A PHYSICAL ADDRESS.
                ; ON ENTRY, HL CONTAINS THE LOGICAL ADDRESS.
                ; ON EXIT, HL CONTAINS THE PHYSICAL ADDRESS.

                ; A LOGICAL ADDRESS IS IN THE FORM (ROW, COLUMN) WITH ROW IN
                ; REG H AND COLUMN IN REG L.  ROW MUST BE IN THE RANGE OF 0
                ; TO NROWS-1 AND COLUMN MUST BE IN THE RANGE OF 0 TO NCOLS-1.

                ; A PHYSICAL ADDRESS IS THE ACTUAL MEMORY ADDRESS IN THE VIDEO
                ; MEMORY.
E0FC D5         GETBA:  PUSH    D
E0FD C5                 PUSH    B
E0FE EB                 XCHG
E0FF 14                 INR     D
E100 21B0FF             LXI     H,-NCOLS
E103 015000             LXI     B,NCOLS
E106 09         GETBA2: DAD     B
E107 15                 DCR     D
E108 C206E1             JNZ     GETBA2
E10B 19                 DAD     D
E10C 1100C0             LXI     D,VIDEO
E10F 19                 DAD     D
E110 C1                 POP     B
E111 D1                 POP     D
E112 C9                 RET

E113                    END
```

.2 INTELLIGENT CRT DRIVER

These notes describe the intelligent CRT software supplied with the VB3 board. The package contains four user-callable routines. They are VB3INIT, VB3IN, VB3STAT, and VB3OUT. VB3INIT is the initialization code and **MUST** be called **BEFORE** calls are made to the other routines. VB3IN and VB3STAT are used to get keystrokes from the keyboard and to check the status of the keyboard (e.g. data available). VB3OUT is the video driver for the VB3 board.

To pass a character to VB3OUT, you must load it into the C register and call VB3OUT. To pass a sequence of characters, VB3OUT **MUST** be called once for each character.

The format type defines the sequence for each control. The three possible formats are:

FORMAT TYPE		FORMAT
a	=	ESC Pn F
b	=	ESC Pn ; Pn F
c	=	ESC Ps F

where ESC is the ASCII code 1BH

Pn is a numeric parameter

Ps is a variable number of selective parameters seperated by semicolons

; is the ASCII code 3BH

F is one of the final characters from the table below

NOTE: Embedded blanks are discarded

For example, to put the cursor at the last column and last row of an 80 x 24 screen, the following ASCII character sequence should be sent:

ESC 80 ; 24 H

In the following table, the parameters that are in parentheses are defaults that will be taken when the actual parameters are omitted.

When no parameters are required, the entire sequence may be abbreviated to a single character (or byte) by adding 80H to the appropriate "final character". For example, to move the cursor backward one position, the single hex byte, 0C4H, could be sent (i.e., no ESC is necessary; the addition of the 080H to the "final character" indicates to VB3OUT that the single character is an entire "ESCape sequence".

Abbreviations used in the following table:
AL = Active Line (containing AP)
AP = Active Position (where the cursor is)
HT = Horizontal Tabulation
VT = Vertical Tabulation

FORMAT TYPE	PARAMS(1)	FINAL CHARACTER ASCII	HEX	MNEMONIC	NAME OF FUNCTION
a	1	@	40	ICH	Insert Character
a	1	A	41	CUU	Cursor Up
a	1	B	42	CUD	Cursor Down
a	1	C	43	DUF	Cursor Forward
a	1	D	44	CUB	Cursor Backward
a	1	E	45	CNL	Cursor Next Line
a	1	F	46	CPL	Cursor Preceding Line
a	1	G	47	CHA	Cursor Horizontal Absolute
b	1;1	H	48	CUP	Cursor Position
a	1	I	49	CHT	Cursor Horizontal Tabulation
c		J	4A	ED	Erase in Display
	0				From AP to end (inclusive)
	1				From start to AP (inclusive)
	2				All of display
c		K	4B	EL	Erase in line
	0				From AP to end (inclusive)
	1				From start to AP (inclusive)
	2				All of line
a	1	L	4C	IL	Insert Line
a	1	M	4D	DL	Delete Line
a	1	P	50	DCH	Delete Character
a	1	S	53	SU	Scroll Up
a	1	T	54	SD	Scroll Down
c		W	57	CTC	Cursor Tabulation Control
	0				Set HT stop at AP
	1				Set VT stop at AP
	2				Clear HT stop at AP
	3				Clear VT stop at AP
	4				Clear all HT stops in AL
	5				Clear all HT stops in device
	6				Clear all VT stops in device
a	1	Y	59	CVT	Cursor Vertical Tabulation
a	1	Z	5A	CBT	Cursor Backward Tabulation
a	1	[	5B	CVA	Cursor Vertical Absolute
c		m	6D	SGR	Select Graphic Rendition
	0				Primary Rendition
	2				Grey scale
	4				Underscore
	5				Blink
	7				Reverse Video
	8				Blank Character
	9				Strike Through
	10				Primary Font (8002)
	11				Thin Graphic Mode
	12				Secondary Font (2716/2732)
	13				Wide Graphic Mode

```
                ;       "ICRT"; Rev 0; Written by Ben L Gee; Dec 9, 1979.

                ;       Modified by Dan Fischler for use with the CP/M
                ;       Assembler-- August 20, 1980

                ;       ICRT is a driver for the SSM VB3 video board.ICRT
                ;       will allow the VB3 to replace most any intelligent
                ;       crt terminal.
                ;
                ;       This software has been used successfully with UCSD
                ;       Pascal and MicroPro Wordstar.


                ;       DEFINE SOME SYSTEM EQUATES
E400 =          ROM     EQU     0E400H          ;Beginning of ROMmable
                                                ;code
0000 =          FALSE   EQU     0
FFFF =          TRUE    EQU     NOT FALSE

                ;       DEFINE THE VB3 PARAMETERS
C000 =          VIDEO   EQU     0C000H  ;Address of video memory
1000 =          OFFSET  EQU     01000H  ;Relative offset to attribute
                                        ;field
00D0 =          VTAC    EQU     0D0H    ;I/O address of CRT Controller
00E0 =          KSTAT   EQU     0E0H    ;Keyboard Status/Board Enable
                                        ;Port
00E1 =          KDATA   EQU     0E1H    ;Keyboard Data/Board Disable Port
0050 =          NCOLS   EQU     80      ;Number of columns
0018 =          NROWS   EQU     24      ;Number of rows
0000 =          SKEW    EQU     0
020D =          SCANF   EQU     525     ;Scans per frame
000B =          SCANR   EQU     11      ;Scans per data row
0071 =          HCOUNT  EQU     113     ;Character times in 1 horiz scan
                                        ;line
FFFF =          INTERL  EQU     TRUE    ;Dispay is interlaced
0005 =          CODE    EQU     5       ;80 columns
0003 =          NORMAL  EQU     3       ;Attribute for alphanumerics
0004 =          REVERS  EQU     4       ;Attribute for reverse video
0008 =          BLANK   EQU     8       ;Attribute for blank video
0010 =          UNDRLN  EQU     16      ;Attribute for underline
0020 =          BLINK   EQU     32      ;Attribute for blinking character
0040 =          STRIKE  EQU     64      ;Attribute for strike-thru
0080 =          REDUCE  EQU     128     ;Attribute for reduced intensity
0020 =          MAXESC  EQU     32      ;Max size of an escape sequence

                ;       DEFINE SOME ASCII VALUES
0008 =          BS      EQU     08H     ;Back space
0009 =          HT      EQU     09H     ;Horizontal Tab
000A =          LF      EQU     0AH     ;Line Feed
000B =          VT      EQU     0BH     ;Vertical Tab
000D =          CR      EQU     0DH     ;Cursor Return
001B =          ESC     EQU     1BH     ;Escape Character
0040 =          ICH     EQU     40H     ;Insert Character
0041 =          CUU     EQU     41H     ;Cursor Up
0042 =          CUD     EQU     42H     ;Cursor Down
```

```
0043 =          CUF     EQU     43H     ;Cursor Forward
0044 =          CUB     EQU     44H     ;Cursor Backward
0045 =          CNL     EQU     45H     ;Cursor Next Line
0046 =          CPL     EQU     46H     ;Cursor Preceding Line
0047 =          CHA     EQU     47H     ;Cursor Horizontal Absolute
0048 =          CUP     EQU     48H     ;Cursor Position
0049 =          CHT     EQU     49H     ;Cursor Horizontal Tab
004A =          ED      EQU     4AH     ;Erase In Display
004B =          EL      EQU     4BH     ;Erase In Line
004C =          IL      EQU     4CH     ;Insert Line
004D =          DL      EQU     4DH     ;Delete Line
0050 =          DCH     EQU     50H     ;Delete Character
0053 =          SU      EQU     53H     ;Scroll Up
0054 =          SD      EQU     54H     ;Scroll Down
0057 =          CTC     EQU     57H     ;Cursor Tab Control
0059 =          CVT     EQU     59H     ;Cursor Vertical Tab
005A =          CBT     EQU     5AH     ;Cursor Backward Tab
005B =          CVA     EQU     5BH     ;Cursor Vertical Absolute
006D =          SGR     EQU     6DH     ;Select Graphic Rendition

                ;       VIDEO DRIVER SUBROUTINES
                ;       ENTRIES POINTS ARE:
                ;
                ;           VBINIT - To initialize the hardware and
                ;                    software
                ;           VBIN   - To get a keystroke from the keyboard
                ;           VBOUT  - To output a character to the display
                ;           VBSTAT - To check the keyboard status
                ;
                ;       CALLING SEQUENCES:
                ;
                ;           CALL  VBINIT
                ;
                ;           CALL  VBIN
                ;           STA   KEYSTROKE     ;Result in Register A
                ;
                ;
                ;           MVI   C,DATA        ;Output character in
                ;           MOV   C,A           ;Register C
                ;           CALL  VBOUT
                ;
                ;           CALL  VBSTAT        ;Returns Register A=0 if
                ;                               ;No keystroke is available
                ;                               ;Else, returns Register
                ;                               ;A=0FFH
                ;

E400                    ORG     ROM
E400 C30CE4             JMP     VBINIT
E403 C359E4             JMP     VBIN
E406 C362E4             JMP     VBOUT
E409 C350E4             JMP     VBSTAT


E40C D3E0       VBINIT: OUT     KSTAT           ;Enable the VB3 Board
```

```
E40E 3E03               MVI     A,NORMAL
E410 3280C7             STA     ATTRIB          ;Set default attribute
E413 3E17               MVI     A,NROWS-1
E415 3283C7             STA     LSTROW          ;Initialize LASTROW
                                                ;to NROWS-1.
E418 AF                 XRA     A
E419 3284C7             STA     MODE            ;Not in Escape mode
E41C CDDEE6             CALL    CTC5            ;Clear all horizontal
                                                ;tab stops
E41F CDEBE6             CALL    CTC6            ;Clear all vertical
                                                ;tab stops
E422 210000             LXI     H,0
E425 CD8AE8             CALL    EEOS            ;Clear the screen
E428 CD05E9             CALL    CURON           ;Put the cursor on the
                                                ;screen

                ; INITALIZE ALL OF THE CONTROL REGISTERS

E42B D3DE               OUT     VTAC+14
E42D D3DA               OUT     VTAC+10         ;Reset the VTAC

E42F 3E70               MVI     A,HCOUNT-1
E431 D3D0               OUT     VTAC+0

                        IF      NOT INTERL
                        MVI     A,03CH
                        ENDIF
                        IF      INTERL
E433 3EBC               MVI     A,0BCH
                        ENDIF
E435 D3D1               OUT     VTAC+1

                        IF      NOT INTERL
                        MVI     A,(SCANR-1)*8+CODE
                        ENDIF
                        IF      INTERL
E437 3E4D               MVI     A,(SCANR-2)*8+CODE
                        ENDIF
E439 D3D2               OUT     VTAC+2

E43B 3E17               MVI     A,SKEW*64+NROWS-1
E43D D3D3               OUT     VTAC+3

                        IF      INTERL
E43F 3E06               MVI     A,(SCANF-513)/2
                        ENDIF
                        IF      NOT INTERL
                        MVI     A,(SCANF-256)/2
                        ENDIF
E441 D3D4               OUT     VTAC+4

E443 3E49               MVI     A,(SCANF-NROWS*SCANR)/4+8
E445 D3D5               OUT     VTAC+5          ;Try to center the data
                                                ;on the screen
```

```
E447 3E17              MVI     A,NROWS-1
E449 D3D6              OUT     VTAC+6

E44B D3DE              OUT     VTAC+14         ;Start the timing chain
E44D D3E1              OUT     KDATA           ;Disable the VB3 board
E44F C9                RET

E450 DBE0      VBSTAT: IN      KSTAT           ;Check keyboard status
E452 2F                CMA
E453 E601              ANI     1
E455 C8                RZ                      ;Return a zero if no data
E456 3EFF              MVI     A,0FFH
E458 C9                RET                     ;Else return a 0FFH

E459 CD50E4    VBIN:   CALL    VBSTAT          ;Return the next key
E45C CA59E4            JZ      VBIN            ;Wait until one is
                                               ;available
E45F DBE1              IN      KDATA           ;Get it
E461 C9                RET                     ;And return

               ; VBOUT IS THE MAIN DISPLAY DRIVER FOR THE VB3 BOARD
E462 E5        VBOUT:  PUSH    H               ;First,save all registers
E463 D5                PUSH    D
E464 C5                PUSH    B
E465 D3E0              OUT     KSTAT           ;And enable the VB3 board
E467 CD1AE9            CALL    CUROFF          ;Then,turn off the cursor
E46A CD76E4            CALL    PROCES          ;And process character
E46D CD05E9            CALL    CURON           ;Now, turn cursor back on
E470 D3E1              OUT     KDATA           ;Disable the VB3 board
E472 C1                POP     B               ;And restore registers
E473 D1                POP     D
E474 E1                POP     H
E475 C9                RET

               ; ON ENTRY,REG HL = LOGICAL CURSOR ADDR. AND REG C = DATA.
E476 3A84C7    PROCES: LDA     MODE            ;See if we received an
                                               ;ESC character
E479 B7                ORA     A
E47A C2B7E4            JNZ     PESCSE          ;JIF we are in an escape
                                               ;sequence
E47D 79                MOV     A,C             ;Move data to register A
E47E FE80              CPI     80H             ;Check for range of 80
                                               ;to FF
E480 D2B4E4            JNC     PESC            ;It is an ESC sequence if
                                               ;>7FH
E483 FE1B              CPI     ESC             ;Check for control
                                               ;characters
E485 CAA7E4            JZ      DOESC           ;If match,then do control
                                               ;character
E488 FE0D              CPI     CR
E48A CA0AE8            JZ      DOCR
E48D FE0A              CPI     LF
E48F CAF8E7            JZ      DOLF
E492 FE09              CPI     HT
E494 CA1BE8            JZ      DOHT
```

```
E497 FE08              CPI     BS
E499 CA0DE8            JZ      DOBS
E49C FE0B              CPI     VT
E49E CA2FE8            JZ      DOVT
E4A1 FE20              CPI     ' '
E4A3 D8                RC                      ;Skip other control
                                               ;characters
E4A4 C3EBE7            JMP     DODAT           ;Must be a data character

E4A7 3E01      DOESC:  MVI     A,1
E4A9 3284C7            STA     MODE            ;We are now reading an
                                               ;ESC sequence
E4AC AF                XRA     A
E4AD 3285C7            STA     COUNT           ;Init the CRT for the
                                               ;sequence
E4B0 3286C7            STA     POINT
E4B3 C9                RET
E4B4 CDA7E4    PESC:   CALL    DOESC
E4B7 3A85C7    PESCSE: LDA     COUNT
E4BA FE20              CPI     MAXESC          ;Check for buffer
                                               ;overflow
E4BC CADAE7            JZ      ABORT
E4BF EB                XCHG                    ;Save HL
E4C0 6F                MOV     L,A             ;Index into buffer
                                               ;goes to HL
E4C1 2600              MVI     H,0
E4C3 3C                INR     A
E4C4 3285C7            STA     COUNT           ;Increment pointer
E4C7 79                MOV     A,C             ;Get data
E4C8 E67F              ANI     7FH             ;7 bits only
E4CA 0187C7            LXI     B,ESCSEQ        ;Get base of buffer
E4CD 09                DAD     B               ;Compute current cell
E4CE 77                MOV     M,A             ;Save data
E4CF EB                XCHG                    ;Restore HL
E4D0 FE40              CPI     '@'             ;Not a terminator if <40H
E4D2 D8                RC
E4D3 4F                MOV     C,A             ;Save the data
E4D4 AF                XRA     A
E4D5 3284C7            STA     MODE            ;ESC seq done, now
                                               ;process it
E4D8 79                MOV     A,C             ;Get the data again
E4D9 FE40              CPI     ICH
E4DB CA4DE5            JZ      DOICH           ;JIF Insert Character
E4DE FE41              CPI     CUU
E4E0 CA5EE5            JZ      DOCUU           ;JIF Cursor Up
E4E3 FE42              CPI     CUD
E4E5 CA6FE5            JZ      DOCUD           ;JIF Cursor Down
E4E8 FE43              CPI     CUF
E4EA CA80E5            JZ      DOCUF           ;JIF Cursor Forward
E4ED FE44              CPI     CUB
E4EF CA91E5            JZ      DOCUB           ;JIF Cursor Back
E4F2 FE45              CPI     CNL
E4F4 CAA2E5            JZ      DOCNL           ;JIF Cursor Next Line
E4F7 FE46              CPI     CPL
E4F9 CAB3E5            JZ      DOCPL           ;JIF Cursor Preceding Lin
```

```
E4FC FE47               CPI     CHA
E4FE CAC4E5             JZ      DOCHA           ;JIF Cursor Horizontal
                                                ;Absolute
E501 FE48               CPI     CUP
E503 CACFE5             JZ      DOCUP           ;JIF Cursor Position
E506 FE49               CPI     CHT
E508 CAD5E5             JZ      DOCHT           ;JIF Cursor Horiz. Tab
E50B FE4A               CPI     ED
E50D CAE6E5             JZ      DOED            ;JIF Erase in Display
E510 FE4B               CPI     EL
E512 CA0EE6             JZ      DOEL            ;JIF Erase in Line
E515 FE4C               CPI     IL
E517 CA35E6             JZ      DOIL            ;JIF Insert Line
E51A FE4D               CPI     DL
E51C CA46E6             JZ      DODL            ;JIF Delete Line
E51F FE50               CPI     DCH
E521 CA57E6             JZ      DODCH           ;JIF Delete Character
E524 FE53               CPI     SU
E526 CA68E6             JZ      DOSU            ;JIF Scroll Up
E529 FE54               CPI     SD
E52B CA79E6             JZ      DOSD            ;JIF Scroll Down
E52E FE57               CPI     CTC
E530 CA8AE6             JZ      DOCTC           ;JIF Cursor Tab Control
E533 FE59               CPI     CVT
E535 CAF8E6             JZ      DOCVT           ;JIF Cursor Vertical Tab
E538 FE5A               CPI     CBT
E53A CA09E7             JZ      DOCBT           ;JIF Cursor Backward Tab
E53D FE5B               CPI     CVA
E53F CA1AE7             JZ      DOCVA           ;JIF Cursor Vertical
                                                ;Absolute
E542 FE6D               CPI     SGR
E544 CA25E7             JZ      DOSGR           ;JIF Select Graphic
                                                ;Rendition
E547 FE7F               CPI     7FH
E549 CA8FE7             JZ      DODEL
E54C C9                 RET                     ;None of the above,
                                                ;so skip it

E54D 3E01       DOICH:  MVI     A,1             ;Insert Character
E54F CDA0E7             CALL    GET1P
E552 C8                 RZ
E553 47                 MOV     B,A
E554 C5         ICH1:   PUSH    B
E555 CD55E9             CALL    INCHAR
E558 C1                 POP     B
E559 05                 DCR     B
E55A C254E5             JNZ     ICH1
E55D C9                 RET

E55E 3E01       DOCUU:  MVI     A,1             ;Cursor Up
E560 CDA0E7             CALL    GET1P           ;Get a parameter,
                                                ;1 is default
E563 C8                 RZ                      ;If value is zero,
                                                ;then done
E564 47                 MOV     B,A
```

```
E565 C5          CUU1:   PUSH    B               ;Repeat till count
                                                 ;is zero
E566 CD58E8              CALL    UP
E569 C1                  POP     B
E56A 05                  DCR     B
E56B C265E5              JNZ     CUU1
E56E C9                  RET

E56F 3E01        DOCUD:  MVI     A,1             ;Cursor Down
E571 CDA0E7              CALL    GET1P
E574 C8                  RZ
E575 47                  MOV     B,A
E576 C5          CUD1:   PUSH    B
E577 CD5FE8              CALL    DOWN
E57A C1                  POP     B
E57B 05                  DCR     B
E57C C276E5              JNZ     CUD1
E57F C9                  RET

E580 3E01        DOCUF:  MVI     A,1
E582 CDA0E7              CALL    GET1P
E585 C8                  RZ
E586 47                  MOV     B,A
E587 C5          CUF1:   PUSH    B
E588 CD6EE8              CALL    RIGHT
E58B C1                  POP     B
E58C 05                  DCR     B
E58D C287E5              JNZ     CUF1
E590 C9                  RET

E591 3E01        DOCUB:  MVI     A,1
E593 CDA0E7              CALL    GET1P
E596 C8                  RZ
E597 47                  MOV     B,A
E598 C5          CUB1:   PUSH    B
E599 CD67E8              CALL    LEFT
E59C C1                  POP     B
E59D 05                  DCR     B
E59E C298E5              JNZ     CUB1
E5A1 C9                  RET

E5A2 3E01        DOCNL:  MVI     A,1             ;Cursor Next Line
E5A4 CDA0E7              CALL    GET1P
E5A7 C8                  RZ
E5A8 47                  MOV     B,A
E5A9 C5          CNL1:   PUSH    B
E5AA CDF6E7              CALL    NEXTLN
E5AD C1                  POP     B
E5AE 05                  DCR     B
E5AF C2A9E5              JNZ     CNL1
E5B2 C9                  RET

E5B3 3E01        DOCPL:  MVI     A,1             ;Cursor Preceding Line
E5B5 CDA0E7              CALL    GET1P
E5B8 C8                  RZ
```

```
E5B9 47                 MOV     B,A
E5BA C5         CPL1:   PUSH    B
E5BB CD76E8             CALL    PREVLN
E5BE C1                 POP     B
E5BF 05                 DCR     B
E5C0 C2BAE5             JNZ     CPL1
E5C3 C9                 RET

E5C4 3E01       DOCHA:  MVI     A,1             ;Cursor Horiz. Absolute
E5C6 CDA0E7             CALL    GET1P
E5C9 3D                 DCR     A
E5CA FE50               CPI     NCOLS
E5CC D0                 RNC
E5CD 6F                 MOV     L,A             ;Put the cursor on
                                                ;that column
E5CE C9                 RET

E5CF CDC4E5     DOCUP:  CALL    DOCHA           ;Cursor position(Goto XY)
E5D2 C31AE7             JMP     DOCVA

E5D5 3E01       DOCHT:  MVI     A,1             ;Cursor Horiz. Tabulation
E5D7 CDA0E7             CALL    GET1P
E5DA C8                 RZ
E5DB 47                 MOV     B,A
E5DC C5         CHT1:   PUSH    B
E5DD CD1BE8             CALL    DOHT
E5E0 C1                 POP     B
E5E1 05                 DCR     B
E5E2 C2DCE5             JNZ     CHT1
E5E5 C9                 RET

E5E6 3E00       DOED:   MVI     A,0             ;Erase in Display
E5E8 CDA0E7             CALL    GET1P
E5EB CA8AE8             JZ      EEOS
E5EE E5                 PUSH    H
E5EF FE01               CPI     1
E5F1 C201E6             JNZ     ED1
E5F4 EB                 XCHG
E5F5 210000             LXI     H,0000H
E5F8 012003             LXI     B,NORMAL*256+' '
E5FB CD9CE8             CALL    FILLS           ;Erase from start
                                                ;to cursor
E5FE C30CE6             JMP     ED2

E601 FE02       ED1:    CPI     2
E603 C20CE6             JNZ     ED2
E606 210000             LXI     H,0
E609 CD8AE8             CALL    EEOS            ;Erase entire display
E60C E1         ED2:    POP     H               ;Restore cursor address
E60D C9                 RET

E60E 3E00       DOEL:   MVI     A,0             ;Erase in Line
E610 CDA0E7             CALL    GET1P
E613 CA93E8             JZ      EEOL            ;Erase from cursor to
                                                ;end of line
```

```
E616 E5                 PUSH    H
E617 FE01               CPI     1
E619 C229E6             JNZ     EL1
E61C EB                 XCHG
E61D 62                 MOV     H,D
E61E 2E00               MVI     L,0
E620 012003             LXI     B,NORMAL*256+' '
E623 CD9CE8             CALL    FILLS           ;Erase from the start of
                                                ;line to cursor
E626 C333E6             JMP     EL2

E629 FE02       EL1:    CPI     2
E62B C233E6             JNZ     EL2
E62E 2E00               MVI     L,0
E630 CD93E8             CALL    EEOL            ;Erase the entire line
E633 E1         EL2:    POP     H
E634 C9                 RET

E635 3E01       DOIL:   MVI     A,1             ;Insert Line
E637 CDA0E7             CALL    GET1P
E63A C8                 RZ
E63B 47                 MOV     B,A
E63C C5         IL1:    PUSH    B
E63D CD22E9             CALL    INSELN
E640 C1                 POP     B
E641 05                 DCR     B
E642 C23CE6             JNZ     IL1
E645 C9                 RET

E646 3E01       DODL:   MVI     A,1             ;Delete Line
E648 CDA0E7             CALL    GET1P
E64B C8                 RZ
E64C 47                 MOV     B,A
E64D C5         DL1:    PUSH    B
E64E CD3DE9             CALL    DELLN
E651 C1                 POP     B
E652 05                 DCR     B
E653 C24DE6             JNZ     DL1
E656 C9                 RET

E657 3E01       DODCH:  MVI     A,1             ;Delete Character
E659 CDA0E7             CALL    GET1P
E65C C8                 RZ
E65D 47                 MOV     B,A
E65E C5         DCH1:   PUSH    B
E65F CD71E9             CALL    DLCHAR
E662 C1                 POP     B
E663 05                 DCR     B
E664 C25EE6             JNZ     DCH1
E667 C9                 RET

E668 3E01       DOSU:   MVI     A,1             ;Scroll Up
E66A CDA0E7             CALL    GET1P
E66D C8                 RZ
E66E 47                 MOV     B,A
```

```
E66F C5          SU1:    PUSH    B
E670 CDE5E8              CALL    RLUP
E673 C1                  POP     B
E674 05                  DCR     B
E675 C26FE6              JNZ     SU1
E678 C9                  RET

E679 3E01        DOSD:   MVI     A,1             ;Scroll Down
E67B CDA0E7              CALL    GET1P
E67E C8                  RZ
E67F 47                  MOV     B,A
E680 C5          SD1:    PUSH    B
E681 CDF5E8              CALL    RLDWN
E684 C1                  POP     B
E685 05                  DCR     B
E686 C280E6              JNZ     SD1
E689 C9                  RET

E68A AF          DOCTC:  XRA     A               ;Cursor Tab Control
E68B CDA0E7      CTCA:   CALL    GET1P
E68E 3C                  INR     A
E68F C8                  RZ
E690 4F                  MOV     C,A
E691 E5                  PUSH    H
E692 CD9BE6              CALL    PCTC
E695 E1                  POP     H
E696 3EFF                MVI     A,255
E698 C38BE6              JMP     CTCA

E69B 0D          PCTC:   DCR     C
E69C CAB8E6              JZ      CTC0
E69F 0D                  DCR     C
E6A0 CAC1E6              JZ      CTC1
E6A3 0D                  DCR     C
E6A4 CACBE6              JZ      CTC2
E6A7 0D                  DCR     C
E6A8 CAD4E6              JZ      CTC3
E6AB 0D                  DCR     C
E6AC CADEE6              JZ      CTC4
E6AF 0D                  DCR     C
E6B0 CADEE6              JZ      CTC5
E6B3 0D                  DCR     C
E6B4 CAEBE6              JZ      CTC6
E6B7 C9                  RET

E6B8 2600        CTC0:   MVI     H,0
E6BA 11A7C7              LXI     D,HTABS
E6BD 19                  DAD     D
E6BE 36FF                MVI     M,255
E6C0 C9                  RET

E6C1 6C          CTC1:   MOV     L,H
E6C2 2600                MVI     H,0
E6C4 11F7C7              LXI     D,VTABS
E6C7 19                  DAD     D
```

```
E6C8 36FF           MVI     M,255
E6CA C9             RET

E6CB 2600     CTC2: MVI     H,0
E6CD 11A7C7         LXI     D,HTABS
E6D0 19             DAD     D
E6D1 3600           MVI     M,0
E6D3 C9             RET

E6D4 6C       CTC3: MOV     L,H
E6D5 2600           MVI     H,0
E6D7 11F7C7         LXI     D,VTABS
E6DA 19             DAD     D
E6DB 3600           MVI     M,0
E6DD C9             RET

              CTC4:
E6DE 21A7C7   CTC5: LXI     H,HTABS
E6E1 0650           MVI     B,NCOLS
E6E3 3600     CTC5A:MVI     M,0
E6E5 23             INX     H
E6E6 05             DCR     B
E6E7 C2E3E6         JNZ     CTC5A
E6EA C9             RET

E6EB 21F7C7   CTC6: LXI     H,VTABS
E6EE 0618           MVI     B,NROWS
E6F0 3600     CTC6A:MVI     M,0
E6F2 23             INX     H
E6F3 05             DCR     B
E6F4 C2F0E6         JNZ     CTC6A
E6F7 C9             RET

E6F8 3E01     DOCVT:MVI     A,1           ;Cursor Vert. Tabulation
E6FA CDA0E7         CALL    GET1P
E6FD C8             RZ
E6FE 47             MOV     B,A
E6FF C5       CVT1: PUSH    B
E700 CD2FE8         CALL    DOVT
E703 C1             POP     B
E704 05             DCR     B
E705 C2FFE6         JNZ     CVT1
E708 C9             RET

E709 3E01     DOCBT:MVI     A,1           ;Cursor Backward Tab.
E70B CDA0E7         CALL    GET1P
E70E C8             RZ
E70F 47             MOV     B,A
E710 C5       CBT1: PUSH    B
E711 CD44E8         CALL    DOBT
E714 C1             POP     B
E715 05             DCR     B
E716 C210E7         JNZ     CBT1
E719 C9             RET
```

```
E71A 3E01      DOCVA:  MVI     A,1             ;Cursor Vertical Absolute
E71C CDA0E7            CALL    GET1P
E71F 3D                DCR     A
E720 FE18              CPI     NROWS
E722 D0                RNC
E723 67                MOV     H,A
E724 C9                RET

E725 AF        DOSGR:  XRA     A               ;Select Graphic Rendition
E726 CDA0E7    SGRA:   CALL    GET1P
E729 4F                MOV     C,A
E72A 3A80C7            LDA     ATTRIB
E72D 0C                INR     C
E72E C8                RZ
E72F CD3AE7            CALL    PSGR
E732 3280C7            STA     ATTRIB
E735 3EFF              MVI     A,255
E737 C326E7            JMP     SGRA

E73A 0D        PSGR:   DCR     C
E73B CA6AE7            JZ      SGR0
E73E 0D                DCR     C
E73F 0D                DCR     C
E740 CA6DE7            JZ      SGR2
E743 0D                DCR     C
E744 0D                DCR     C
E745 CA70E7            JZ      SGR4
E748 0D                DCR     C
E749 CA73E7            JZ      SGR5
E74C 0D                DCR     C
E74D 0D                DCR     C
E74E CA76E7            JZ      SGR7
E751 0D                DCR     C
E752 CA79E7            JZ      SGR8
E755 0D                DCR     C
E756 CA7CE7            JZ      SGR9
E759 0D                DCR     C
E75A CA7FE7            JZ      SGR10
E75D 0D                DCR     C
E75E CA82E7            JZ      SGR11
E761 0D                DCR     C
E762 CA87E7            JZ      SGR12
E765 0D                DCR     C
E766 CA8CE7            JZ      SGR13
E769 C9                RET

E76A 3E03      SGR0:   MVI     A,3
E76C C9                RET

E76D F680      SGR2:   ORI     REDUCE
E76F C9                RET

E770 F610      SGR4:   ORI     UNDRLN
E772 C9                RET
```

```
E773 F620       SGR5:   ORI     BLINK
E775 C9                 RET

E776 F604       SGR7:   ORI     REVERS
E778 C9                 RET

E779 F608       SGR8:   ORI     BLANK
E77B C9                 RET

E77C F640       SGR9:   ORI     STRIKE
E77E C9                 RET

E77F F603       SGR10:  ORI     3
E781 C9                 RET

E782 E6FC       SGR11:  ANI     0FCH
E784 F602               ORI     2
E786 C9                 RET

E787 E6FC       SGR12:  ANI     0FCH
E789 F601               ORI     1
E78B C9                 RET

E78C E6FC       SGR13:  ANI     0FCH
E78E C9                 RET

E78F E5         DODEL:  PUSH    H
E790 2100C0             LXI     H,VIDEO
E793 018007             LXI     B,NROWS*NCOLS
E796 75         DODEL1: MOV     M,L
E797 23                 INX     H
E798 0B                 DCX     B
E799 79                 MOV     A,C
E79A B0                 ORA     B
E79B 05                 DCR     B
E79C C296E7             JNZ     DODEL1
E79F C9                 RET
E7A0 57         GETIP:  MOV     D,A             ;Save the default value
E7A1 1E00               MVI     E,0             ;Zero the accumulator
E7A3 0186C7             LXI     B,ESCSEQ-1      ;Point to the esc. string
E7A6 E5                 PUSH    H               ;Save the cursor address

                MORE:
E7A7 3A86C7     NOGOOD: LDA     POINT
E7AA 3C                 INR     A               ;Compute the index for
                                                ;the next byte
E7AB 2185C7             LXI     H,COUNT
E7AE BE                 CMP     M               ;See if there is any more
E7AF CAD6E7             JZ      NOMORE          ;JIF None
E7B2 3286C7             STA     POINT           ;Update the pointer
E7B5 6F                 MOV     L,A
E7B6 2600               MVI     H,0
E7B8 09                 DAD     B               ;Point to the next char
                                                ;to come out
E7B9 7E                 MOV     A,M             ;Get it
```

```
E7BA FE3B               CPI     ';'             ;Check for a seperator
E7BC CAD6E7             JZ      NOMORE          ;JIF number is done
E7BF D630               SUI     '0'
E7C1 DAA7E7             JC      NOGOOD
E7C4 FE0A               CPI     10
E7C6 D2A7E7             JNC     NOGOOD          ;JIF char is not from
                                                ;0 to 9
E7C9 57                 MOV     D,A             ;Save the value
E7CA 7B                 MOV     A,E             ;Multiply the accumulated
                                                ;value by 10
E7CB 87                 ADD     A               ;*2
E7CC 5F                 MOV     E,A
E7CD 87                 ADD     A               ;*4
E7CE 87                 ADD     A               ;*8
E7CF 83                 ADD     E               ;*10  Done!
E7D0 82                 ADD     D               ;Add in new digit
E7D1 5F                 MOV     E,A             ;Return it to the
                                                ;accumulator
E7D2 57                 MOV     D,A             ;Wipe out old value
E7D3 C3A7E7             JMP     MORE            ;Go for the next digit

E7D6 7A         NOMORE: MOV     A,D             ;Get the value
E7D7 B7                 ORA     A               ;Set flags
E7D8 E1                 POP     H               ;Restore cursor address
E7D9 C9                 RET

E7DA AF         ABORT:  XRA     A
E7DB 3284C7             STA     MODE    ;Reset the escape flag
E7DE C9                 RET

                ; PUT A DATA CHAR ON THE SCREEN AND ADVANCE THE CURSOR.

E7DF E5         PUTCHAR:PUSH    H               ;Save the logical cursor
                                                ;address
E7E0 CDDAE9             CALL    GETBA           ;Get the physical cursor
                                                ;address
E7E3 110010             LXI     D,OFFSET
E7E6 71                 MOV     M,C             ;Put the data there
E7E7 19                 DAD     D               ;Compute the address of
                                                ;the governing attribute
E7E8 70                 MOV     M,B             ;Put the attribute here
E7E9 E1                 POP     H               ;Get the logical cursor
                                                ;address back
E7EA C9                 RET

E7EB 3A80C7     DODAT:  LDA     ATTRIB          ;Get the current attribute
E7EE 47                 MOV     B,A             ;Set it up for PUTCHAR
E7EF CDDFE7             CALL    PUTCHAR
E7F2 CD6EE8             CALL    RIGHT           ;Move the cursor right
E7F5 C0                 RNZ                     ;Done if the cursor
                                                ;didn't wrap around
                                                ;Else, we must move the
                                                ;cursor to the next line

                ; PUT THE CURSOR ON COLUMN 1 OF THE NEXT LINE.  SCROLL
```

```
                ; THE SCREEN IF THE CURSOR WAS ON THE LAST LINE.

E7F6 2E00       NEXTLN: MVI     L,0             ;Put the cursor on
                                                ;column 1 of the next line
                                                ;And do a line feed

                ; MOVE THE CURSOR DOWN ONE LINE.  SCROLL THE SCREEN
                ; IF THE CURSOR WAS ON THE LAST LINE.

E7F8 CD5FE8     DOLF:   CALL    DOWN
E7FB C0                 RNZ
E7FC 2617               MVI     H,NROWS-1
E7FE E5                 PUSH    H
E7FF CDE5E8             CALL    RLUP
E802 210017             LXI     H,(NROWS-1)*256
E805 CD93E8             CALL    EEOL
E808 E1                 POP     H
E809 C9                 RET

                ; MOVE THE CURSOR TO COL 1 OF THE CURRENT LINE.

E80A 2E00       DOCR:   MVI     L,0
E80C C9                 RET

                ; BACK THE CURSOR BY ONE POSITION.

E80D CD67E8     DOBS:   CALL    LEFT
E810 CC58E8             CZ      UP
E813 E5                 PUSH    H
E814 CDDAE9             CALL    GETBA
E817 3620               MVI     M,' '
E819 E1                 POP     H
E81A C9                 RET

                ; DO A HORIZONTAL TABULATION

E81B 4D         DOHT:   MOV     C,L             ;Save the column number
E81C 11A7C7             LXI     D,HTABS         ;Point to the horizontal
                                                ;tab table
E81F CD6EE8     DOHT1:  CALL    RIGHT           ;Move right
E822 E5                 PUSH    H
E823 AF                 XRA     A               ;Get a zero
E824 67                 MOV     H,A
E825 19                 DAD     D               ;Point to HTABS[COL]
E826 B6                 ORA     M               ;See if true
E827 E1                 POP     H
E828 C0                 RNZ                     ;Done if found a tab stop
E829 7D                 MOV     A,L
E82A B9                 CMP     C
E82B C8                 RZ                      ;Done if no tab stops
E82C C31FE8             JMP     DOHT1           ;Continue looking for
                                                ;tab stops

                ; DO A VERTICAL TABULATION
```

```
E82F 4C        DOVT:   MOV     C,H             ;Save the row number
E830 11F7C7            LXI     D,VTABS         ;Point to the vertical
                                               ;tab table
E833 CD5FE8    DOVT1:  CALL    DOWN            ;Move down
E836 E5                PUSH    H
E837 AF                XRA     A               ;Get a zero
E838 6C                MOV     L,H
E839 67                MOV     H,A
E83A 19                DAD     D               ;Point to VTABS[ROW]
E83B B6                ORA     M               ;See if true
E83C E1                POP     H
E83D C0                RNZ                     ;Done if found a tab stop
E83E 7C                MOV     A,H
E83F B9                CMP     C
E840 C8                RZ                      ;Done if not tab stops
E841 C333E8            JMP     DOVT1           ;Continue looking for
                                               ;tab stops

                ; DO A BACKWARD TABULATION

E844 4D        DOBT:   MOV     C,L             ;Save the column number
E845 11A7C7            LXI     D,HTABS         ;Point to the horizontal
                                               ;tab table
E848 CD67E8    DOBT1:  CALL    LEFT            ;Move left
E84B E5                PUSH    H
E84C AF                XRA     A               ;Get a zero
E84D 67                MOV     H,A
E84E 19                DAD     D               ;Point to HTABS[COL]
E84F B6                ORA     M               ;See if true
3850 E1                POP     H
E851 C0                RNZ                     ;Done if found a tab stop
E852 7D                MOV     A,L
E853 B9                CMP     C
E854 C8                RZ                      ;Done if no tab stops
E855 C348E8            JMP     DOBT1           ;Continue looking for tab
                                               ;stops

                ; MOVE THE CURSOR UP.

E858 7C        UP:     MOV     A,H
E859 25                DCR     H
E85A B7                ORA     A
E85B C0                RNZ
E85C 2617              MVI     H,NROWS-1
E85E C9                RET

                ; MOVE THE CURSOR DOWN.

E85F 24        DOWN:   INR     H
E860 3E18              MVI     A,NROWS
E862 BC                CMP     H
E863 C0                RNZ
E864 2600              MVI     H,0
E866 C9                RET
```

```
                ; MOVE THE CURSOR LEFT.

E867 7D         LEFT:   MOV     A,L
E868 2D                 DCR     L
E869 B7                 ORA     A
E86A C0                 RNZ
E86B 2E4F               MVI     L,NCOLS-1
E86D C9                 RET

                ; MOVE THE CURSOR RIGHT.

E86E 2C         RIGHT:  INR     L
E86F 3E50               MVI     A,NCOLS
E871 BD                 CMP     L
E872 C0                 RNZ
E873 2E00               MVI     L,0
E875 C9                 RET

                ; MOVE THE CURSOR TO THE BEGINNING OF THE PREVIOUS LINE

E876 2E00       PREVLN: MVI     L,0
E878 CD58E8             CALL    UP
E87B C0                 RNZ
E87C 2600               MVI     H,0
E87E E5                 PUSH    H
E87F CDF5E8             CALL    RLDWN
E882 210000             LXI     H,0
E885 CD93E8             CALL    EEOL
E888 E1                 POP     H
E889 C9                 RET

                ; ERASE FROM THE CURSOR POSITION TO THE END OF THE SCREEN

E88A 114F17     EEOS:   LXI     D,((NROWS-1)*256)+NCOLS-1
E88D 012003             LXI     B,NORMAL*256+' '
E890 C39CE8             JMP     FILLS

                ; ERASE FROM THE CURSOR TO THE END OF THE CURRENT LINE.

E893 54         EEOL:   MOV     D,H
E894 1E4F               MVI     E,NCOLS-1
E896 012003             LXI     B,NORMAL*256+' '
E899 C39CE8             JMP     FILLS

                ; FILL THE SCREEN WITH THE DATA IN REG C
                ; THE ATTRIB IN REG B
                ; FROM X,Y LOCATION HL TO DE.

E89C E5         FILLS:  PUSH    H
E89D CDDAE9             CALL    GETBA           ;Get address of start
E8A0 EB                 XCHG
E8A1 CDDAE9             CALL    GETBA           ;Get address of finish
E8A4 7C                 MOV     A,H             ;Compare finish and start
E8A5 BA                 CMP     D
E8A6 C2ABE8             JNZ     CMPD1
```

```
E8A9 7D                 MOV     A,L
E8AA BB                 CMP     D+1
E8AB EB         CMPD1:  XCHG
E8AC D2BCE8             JNC     FILLS1          ;JIF area to fill does
                                                ;not wrap
E8AF D5                 PUSH    D               ;Save finish address
E8B0 C5                 PUSH    B
E8B1 117FC7             LXI     D,VIDEO+NROWS*((NCOLS+15) AND 0FFF0H)-1
E8B4 CDC1E8             CALL    FILLS2          ;Fill from start to end
                                                ;of screen
E8B7 C1                 POP     B
E8B8 D1                 POP     D
E8B9 2100C0             LXI     H,VIDEO         ;Fill from start of
                                                ;screen to finish address
E8BC CDC1E8     FILLS1: CALL    FILLS2          ;Fill the screen
E8BF E1                 POP     H
E8C0 C9                 RET

E8C1 C5         FILLS2: PUSH    B
E8C2 E5                 PUSH    H
E8C3 D5                 PUSH    D
E8C4 CDD5E8             CALL    FILL            ;Fill in the data
E8C7 E1                 POP     H
E8C8 D1                 POP     D
E8C9 010010             LXI     B,OFFSET
E8CC 09                 DAD     B
E8CD EB                 XCHG
E8CE 09                 DAD     B
E8CF C1                 POP     B
E8D0 48                 MOV     C,B
E8D1 CDD5E8             CALL    FILL            ;Fill in the attributes
E8D4 C9                 RET

E8D5 71         FILL:   MOV     M,C
E8D6 23                 INX     H
E8D7 7C                 MOV     A,H
E8D8 BA                 CMP     D
E8D9 C2DEE8             JNZ     CMPD2
E8DC 7D                 MOV     A,L
E8DD BB                 CMP     D+1
E8DE DAD5E8     CMPD2:  JC      FILL
E8E1 CAD5E8             JZ      FILL
E8E4 C9                 RET

                ; SCROLL THE SCREEN UP BY ONE LINE.

E8E5 3A83C7     RLUP:   LDA     LSTROW
E8E8 3C                 INR     A
E8E9 FE18               CPI     NROWS
E8EB C2EFE8             JNZ     RLUP1
E8EE AF                 XRA     A
                RLUP1:
E8EF D3D6               OUT     VTAC+6
E8F1 3283C7             STA     LSTROW
E8F4 C9                 RET
```

```
                ; SCROLL THE SCREEN DOWN ONE LINE

E8F5 3A83C7     RLDWN:  LDA     LSTROW
E8F8 B7                 ORA     A
E8F9 C2FEE8             JNZ     RLDWN1
E8FC 3E18               MVI     A,NROWS
E8FE 3D         RLDWN1: DCR     A
E8FF D3D6               OUT     VTAC+6
E901 3283C7             STA     LSTROW
E904 C9                 RET


                ; TURN THE CURSOR ON.
                ; THE LOGICAL ADDRESS OF THE CURSOR MUST BE IN REG HL.

E905 2281C7     CURON:  SHLD    CURADR
E908 7D                 MOV     A,L
E909 D3DC               OUT     VTAC+12
E90B 3A83C7             LDA     LSTROW
E90E 3C                 INR     A
E90F 84                 ADD     H
E910 FE18               CPI     NROWS
E912 DA17E9             JC      CURON1
E915 D618               SUI     NROWS
E917 D3DD       CURON1: OUT     VTAC+13
E919 C9                 RET

                ; TURN THE CURSOR OFF.
                ; THE LOGICAL ADDRESS OF THE CURSOR IS RETURNED IN REG HL

E91A 2A81C7     CUROFF: LHLD    CURADR
E91D 3EFF               MVI     A,0FFH
E91F D3DC               OUT     VTAC+12
E921 C9                 RET

                ;  THE INTERNAL DESCRIPTION OF THE NEXT FOUR PROCEDURES
                ;  ARE THEIR PASCAL COUNTERPARTS.  THE PROCEDURES WERE
                ;  FIRST WRITTEN IN PASCAL AND THEN TRANSLATED INTO
                ;  ASSEMBLY.  IN EACH CASE I HAVE ASSUMED THAT THE
                ;  FOLLOWING DECLARATIONS HAVE BEEN MADE.


                ;   CONST
                ;       NCOLS = 80; (* OR WHATEVER *)
                ;       NROWS = 24; (* OR WHATEVER *)
                ;
                ;   TYPE
                ;       COLS = 0..NCOLS-1;
                ;       ROWS = 0..NROWS-1;
                ;       LINE = RECORD OF
                ;                 COL : ARRAY[COLS] OF CHAR
                ;              END;
                ;       SCREEN = ARRAY [ROWS] OF LINE;
```

```
                ;
                ;   VAR
                ;       R : ROWS;         (*THE CURRENT CURSOR ROW ADDRESS*)
                ;       C : COLS;         (*THE CURRENT CURSOR COLUMN ADDR.*)
                ;       ROW : SCREEN;     (*THIS IS THE VB3 MEMORY MAPPED SCREEN*)
                ;       RI : ROWS;        (*A COUNTER FOR ROWS*)
                ;       CI : COLS;        (*A COUNTER FOR COLUMNS*)
                ;       BLANK_LINE : LINE;      (* ASSUMED TO BE A BLANK LINE *)

                ;  TO REFERENCE A CHARACTER ON THE SCREEN, SAY:
                ;
                ;               ROW[RI].COL[CI]
                ;
                ;  FOR SOME VALUES OF RI AND CI.
                ;
                ;  TO REFERENCE A ENTIRE LINE OF THE SCREEN, SAY:
                ;
                ;               ROW[RI]
                ;
                ;  FOR SOME VALUE OF RI.



                ;   PROCEDURE INSELN;
                ;     BEGIN
                ;       FOR RI := NROWS-1 TO R+1 DO
                ;         ROW[RI] := ROW[RI-1];
                ;       ROW[R] := BLANK_LINE;
                ;     END;

E922 E5         INSELN: PUSH    H
E923 4C                 MOV     C,H
E924 0C                 INR     C
E925 210017             LXI     H,(NROWS-1)*256
E928 110017             LXI     D,(NROWS-1)*256
E92B 7C         ILLOOP: MOV     A,H
E92C B9                 CMP     C
E92D DA38E9             JC      ILEXIT
E930 25                 DCR     H
E931 CD8AE9             CALL    MVLINE
E934 15                 DCR     D
E935 C32BE9             JMP     ILLOOP

E938 CD93E8     ILEXIT: CALL    EEOL
E93B E1                 POP     H
E93C C9                 RET

                ;   PROCEDURE DELLN;
                ;     BEGIN
                ;       FOR RI := R TO NROWS-2 DO
                ;         ROW[RI] := ROW[RI+1];
                ;       ROW[NROWS-1] := BLANK_LINE;
                ;     END;
```

```
E93D E5         DELLN:  PUSH    H
E93E 2E00               MVI     L,0
E940 54                 MOV     D,H
E941 5D                 MOV     E,L
E942 3E16       DLLOOP: MVI     A,NROWS-2
E944 BA                 CMP     D
E945 DA50E9             JC      DLEXIT
E948 24                 INR     H
E949 CD8AE9             CALL    MVLINE
E94C 14                 INR     D
E94D C342E9             JMP     DLLOOP

E950 CD93E8     DLEXIT: CALL    EEOL
E953 E1                 POP     H
E954 C9                 RET

                ;   PROCEDURE INCHAR;
                ;     BEGIN
                ;       FOR CI := NCOLS-1 DOWNTO C+1 DO
                ;         ROW[R].COL[CI] := ROW[R].COL[CI-1];
                ;       ROW[R].COL[C] := ' ';
                ;       END;

E955 E5         INCHAR: PUSH    H
E956 4D                 MOV     C,L
E957 0C                 INR     C
E958 2E4F               MVI     L,NCOLS-1
E95A 54                 MOV     D,H
E95B 5D                 MOV     E,L
E95C 7D         ICHLOP: MOV     A,L
E95D B9                 CMP     C
E95E DA69E9             JC      ICHEXT
E961 2D                 DCR     L
E962 CDC2E9             CALL    MVCHAR
E965 1D                 DCR     E
E966 C35CE9             JMP     ICHLOP

E969 012003     ICHEXT: LXI     B,NORMAL*256+' '
E96C CDDFE7             CALL    PUTCHAR
E96F E1                 POP     H
E970 C9                 RET

                ;   PROCEDURE DLCHAR;
                ;     BEGIN
                ;       FOR CI := C TO NCOLS-2 DO
                ;         ROW[R].COL[CI] := ROW[R].COL[CI+1];
                ;       ROW[R].COL[NCOLS-1] := ' ';
                ;     END;

E971 E5         DLCHAR: PUSH    H
E972 54                 MOV     D,H
E973 5D                 MOV     E,L
E974 3E4E       DCHLOP: MVI     A,NCOLS-2
E976 BB                 CMP     E
```

```
E977 DA82E9            JC      DCHTXT
E97A 2C                INR     L
E97B CDC2E9            CALL    MVCHAR
E97E 1C                INR     E
E97F C374E9            JMP     DCHLOP

E982 012003    DCHTXT: LXI     B,NORMAL*256+' '
E985 CDDFE7            CALL    PUTCHAR
E988 E1                POP     H
E989 C9                RET

E98A E5        MVLINE: PUSH    H
E98B D5                PUSH    D
E98C C5                PUSH    B
E98D 010010            LXI     B,OFFSET
E990 CDDAE9            CALL    GETBA
E993 E5                PUSH    H
E994 09                DAD     B
E995 EB                XCHG
E996 CDDAE9            CALL    GETBA
E999 E5                PUSH    H
E99A 09                DAD     B
E99B EB                XCHG
E99C 015000            LXI     B,NCOLS
E99F 04                INR     B
E9A0 7E        HERE1:  MOV     A,M
E9A1 12                STAX    D
E9A2 23                INX     H
E9A3 13                INX     D
E9A4 0D                DCR     C
E9A5 C2A0E9            JNZ     HERE1
E9A8 05                DCR     B
E9A9 C2A0E9            JNZ     HERE1
E9AC D1                POP     D
E9AD E1                POP     H
E9AE 015000            LXI     B,NCOLS
E9B1 04                INR     B
E9B2 7E        HERE2:  MOV     A,M
E9B3 12                STAX    D
E9B4 23                INX     H
E9B5 13                INX     D
E9B6 0D                DCR     C
E9B7 C2B2E9            JNZ     HERE2
E9BA 05                DCR     B
E9BB C2B2E9            JNZ     HERE2
E9BE C1                POP     B
E9BF D1                POP     D
E9C0 E1                POP     H
E9C1 C9                RET

E9C2 E5        MVCHAR: PUSH    H
E9C3 D5                PUSH    D
E9C4 C5                PUSH    B
E9C5 CDDAE9            CALL    GETBA
E9C8 EB                XCHG
```

```
E9C9 CDDAE9             CALL    GETBA
E9CC 1A                 LDAX    D
E9CD 77                 MOV     M,A
E9CE 010010             LXI     B,OFFSET
E9D1 09                 DAD     B
E9D2 EB                 XCHG
E9D3 09                 DAD     B
E9D4 7E                 MOV     A,M
E9D5 12                 STAX    D
E9D6 C1                 POP     B
E9D7 D1                 POP     D
E9D8 E1                 POP     H
E9D9 C9                 RET

                ; CONVERT A LOGICAL ADDRESS TO A PHYSICAL ADDRESS.
                ; ON ENTRY, HL CONTAINS THE LOGICAL ADDRESS.
                ; ON EXIT, HL CONTAINS THE PHYSICAL ADDRESS.

                ; A LOGICAL ADDRESS IS IN THE FORM (ROW, COLUMN) WITH ROW
                ; IN REG H AND COLUMN IN REG L.  ROW MUST BE IN THE RANGE
                ; OF 0 TO NROWS-1 AND COLUMN MUST BE IN THE RANGE OF 0 TO
                ; NCOLS-1.

                ; A PHYSICAL ADDRESS IS THE ACTUAL MEMORY ADDRESS IN THE
                ; VIDEO MEMORY.
E9DA C5         GETBA:  PUSH    B
E9DB 3A83C7             LDA     LSTROW
E9DE 3C                 INR     A
E9DF 84                 ADD     H
E9E0 FE18               CPI     NROWS
E9E2 DAE7E9             JC      GETBA1
E9E5 D618               SUI     NROWS
E9E7 4F         GETBA1: MOV     C,A
E9E8 0600               MVI     B,0
E9EA 7D                 MOV     A,L
E9EB 21F8E9             LXI     H,MAPPER
E9EE 09                 DAD     B
E9EF 09                 DAD     B
E9F0 4F                 MOV     C,A
E9F1 7E                 MOV     A,M
E9F2 23                 INX     H
E9F3 66                 MOV     H,M
E9F4 6F                 MOV     L,A
E9F5 09                 DAD     B
E9F6 C1                 POP     B
E9F7 C9                 RET

                ;       The following section (MAPPER) saves space for a
                ;       mapping table.
                ;       The number of DW's should be equal to the number
                ;       of lines.

                MAPPER:
                J       SET     (NCOLS+15) AND 0FFF0H
E9F8 00C0               DW      VIDEO+J*0
```

```
E9FA 50C0               DW      VIDEO+J*1
E9FC A0C0               DW      VIDEO+J*2
E9FE F0C0               DW      VIDEO+J*3
EA00 40C1               DW      VIDEO+J*4
EA02 90C1               DW      VIDEO+J*5
EA04 E0C1               DW      VIDEO+J*6
EA06 30C2               DW      VIDEO+J*7
EA08 80C2               DW      VIDEO+J*8
EA0A D0C2               DW      VIDEO+J*9
EA0C 20C3               DW      VIDEO+J*10
EA0E 70C3               DW      VIDEO+J*11
EA10 C0C3               DW      VIDEO+J*12
EA12 10C4               DW      VIDEO+J*13
EA14 60C4               DW      VIDEO+J*14
EA16 B0C4               DW      VIDEO+J*15
EA18 00C5               DW      VIDEO+J*16
EA1A 50C5               DW      VIDEO+J*17
EA1C A0C5               DW      VIDEO+J*18
EA1E F0C5               DW      VIDEO+J*19
EA20 40C6               DW      VIDEO+J*20
EA22 90C6               DW      VIDEO+J*21
EA24 E0C6               DW      VIDEO+J*22
EA26 30C7               DW      VIDEO+J*23
EA28 80C7               DW      VIDEO+J*24

EA2A =          ENDROM  EQU     $

C780                    ORG     VIDEO+NCOLS*NROWS
C780            ATTRIB: DS      1       ;THE CURRENT SCREEN ATTRIB
C781            CURADR: DS      2       ;THE LOGICAL CURSOR ADDRESS
C783            LSTROW: DS      1       ;THE LAST ROW DISPLAYED ON THE
                                        ;SCREEN
C784            MODE:   DS      1       ;BOOLEAN, TRUE IF RECEIVING AN
                                        ;ESCAPE SEQUENCE
C785            COUNT:  DS      1       ;INDEX INTO ESCSEQ
C786            POINT:  DS      1       ;SAME AS ABOVE EXCEPT USED FOR
                                        ;READING
C787            ESCSEQ: DS      MAXESC  ;HOLDS THE ESCAPE SEQUENCE
C7A7            HTABS:  DS      NCOLS   ;USED FOR SETTING HORIZONTAL TABS
C7F7            VTABS:  DS      NROWS   ;SAME FOR VERTICAL TABS

C80F =          ENDRAM  EQU     $

EA2A                    ORG     ENDROM          ;This address will appear
                                                ;on the console after the
                                                ;assembly

EA2A                    END     ROM
```

6.3 VB3 GRAPHICS ROUTINE

Adapted from the SSM VB1C graphics routine, this routine treats the VB3 as a matrix of "160 across" by "4 times the number of rows down". The area of the VB3 display that the user uses with this routine must be set to the graphics mode with the correct bit set in the attribute byte (see section 5.7). The H and L registers are used to specify each particular XY coordinate of the matrix (refer to the comments in the routine).

```
;VB3 GRAPHICS ROUTINE.
;WRITTEN BY MALCOLM WRIGHT, 2-3-80
;CONCEPTS FROM CAL OHME'S VB1C ROUTINE.

;ENTRY CONDITIONS
;       H=VERTICAL COORDINATE
;       L=HORIZONTAL COORDINATE
;EXIT CONDITIONS
;       A=USED
;       B=PRESERVED
;       C=BIT MASK FOR DOT
;       D&E=MEMORY ADDRESS OF DOT
;       H=VERTICAL COORDINATE NORMALIZED
;       L=HORIZONTAL COORDINATE NORMALIZED

;SET UP FOR 80 CHARACTERS ACROSS TIMES 2.
;SET FOR "NROW" OF ROWS DOWN.

CHECK:  CALL    CNVRT   ;ZERO FLAG=1 IF WHITE
                        ;ZERO FLAG=0 IF BLACK
        ANA     C
        RET
WHITE:  CALL    CNVRT   ;CHANGE TO WHITE
        ORA     C       ;ADD DOT
        STAX    D
        RET
BLACK:  CALL    CNVRT   ;CHANGE TO BLACK
        ORA     C       ;ADD DOT
        XRA     C       ;MAKE BLACK
        STAX    D
        RET

;CONVERT H&L AS Y&X POSITIONS INTO ACTUAL MEMORY ADDRESS.
CNVRT:  PUSH    B       ;SAVE B&C
        ;NORMALIZE X-AXIS
        MOV     A,L
        CPI     0FFH    ;BELOW LIMIT, NEG.
        JNZ     X1
        MVI     A,(80*2)-1 ;RESET TO RIGHT SIDE
X1:     CPI     80*2    ;ABOVE UPPER LIMIT
        JC      X2
        XRA     A       ;RESET TO LEFT SIDE
X2:     MOV     L,A
        ;MORMALIZE Y-AXIS
        MOV     A,H
        CPI     0FFH    ;BELOW LIMIT, NEG.
        JNZ     Y1
        MVI     A,(NROW*4)-1 ;RESET TO TOP
Y1:     CPI     NROW*4  ;ABOVE UPPER LIMIT
        JC      Y2
        XRA     A       ;RESET TO BOTTOM
Y2:     MOV     H,A
        ;REVERSE(FLIP-OVER) Y-AXIS
        PUSH    H       ;SAVE H&L NORMALIZED
        MOV     C,A
```

```
MVI     A,(NROW*4)-1
SUB     C       ;FLIP OVER
;FIND ACTUAL DELTA HEIGHT
MVI     C,0     ;CLEAR MASK INDEX
ORA     A       ;CLEAR CARRY
RAR             ;DIVIDE BY 2
MOV     B,A     ;SAVE QUOTIENT
MOV     A,C
RAL             ;UPDATE INDEX
MOV     C,A
MOV     A,B     ;GET QUOTIENT
ORA     A       ;CLEAR CARRY
RAR             ;DIVIDE BY 2
MOV     E,A     ;SAVE DELTA HEIGHT
MOV     A,C
RAL             ;UPDATE INDEX
MOV     C,A     ;SAVE MASK INDEX(REMAINDER)
;COMPUTE VERTICAL ROW TIMES 80
MVI     D,0
MOV     L,E     ;L = QUOTIENT
MOV     H,D
DAD     H       ;X2
DAD     H       ;X4
DAD     D       ;X5
DAD     H       ;X10
DAD     H       ;X20
DAD     H       ;X40
DAD     H       ;X80
XCHG            ;SAVE 80*NROW
POP     H
;FIND ACTUAL WIDTH DELTA
PUSH    H
ORA     A       ;CLEAR CARRY
MOV     A,L     ;X-AXIS
RAR             ;DIVIDE BY 2
MOV     L,A
MOV     A,C
RAL             ;UPDATE MASK INDEX
MOV     C,A
;CREATE ACTUAL MEMORY ADDRESS FOR PIXEL
MVI     H,0     ;H&L= WIDTH
DAD     D       ;ADD TO HEIGHT FOR OFFSET
LXI     D,VID   ;GET VB3 ADDRESS
DAD     D       ;OFFSET+BASE ADDRESS
XCHG            ;D&E= ACTUAL MEMORY ADDR.
;GET BIT MASK
MVI     B,0     ;B&C= MASK INDEX
LXI     H,BTBL  ;START OF MASK TABLE
DAD     B       ;ADD INDEX
MOV     A,M     ;GET MASK
;RETURN AND GET VB3 BYTE
POP     H
POP     B
MOV     C,A     ;C=MASK
LDAX    D
```

```
        RET

;MASK BIT TABLE(2 X 4)
BTBL:   DB      80H
        DB      8
        DB      20H
        DB      2
        DB      40H
        DB      4
        DB      10H
        DB      1

        END
```

6.4 MEMORY TEST

```
                ;       Simple Memory Test
                ;       Written by Andrew Schneider
                ;       Modified by Malcolm Wright
                ;       Copyright 1977 by SSM

                ;       Set "START" to the starting address of
                ;       memory to be tested. Set "MEND" to the last
                ;       address of memory to be checked.

                ;       The program will stop (HALT) when complete
                ;       or if an error was found. "GORB" (good or
                ;       bad) will be set to 00H for good memory or
                ;       to the byte pattern that would not read or
                ;       write correctly into memory. "LAST" is the
                ;       location where the last address tested will
                ;       be saved. If memory is good, then LAST=MEND.

0100 =          BEGIN   EQU     0100H   ;Start of program
C000 =          START   EQU     0C000H  ;Beginning address
C7FF =          MEND    EQU     0C7FFH  ;Ending address

0100                    ORG     BEGIN
0100 2100C0             LXI     H,START
0103 11FFC7             LXI     D,MEND
0106 2B                 DCX     H
0107 23         LOOP:   INX     H
0108 3E7F               MVI     A,7FH
010A 07         CHECK:  RLC
010B 77                 MOV     M,A
010C BE                 CMP     M
010D C22001             JNZ     ERROR
0110 B7                 ORA     A
0111 FA0A01             JM      CHECK
0114 7B                 MOV     A,E
0115 BD                 CMP     L
0116 C20701             JNZ     LOOP
0119 7A                 MOV     A,D
011A BC                 CMP     H
011B C20701             JNZ     LOOP
011E 3E00               MVI     A,0
0120 322701     ERROR:  STA     GORB    ;If using an IMSAI front panel
                                        ;replace with       CMA
                                        ;                   OUT  0FFH
                                        ;to display byte on front panel.
0123 222801             SHLD    LAST
0126 76                 HLT
0127 00         GORB:   DB      0
0128 0000       LAST:   DW      0
012A                    END
```

6.5 VIDEO TEST

```
                ; VB3 VIDEO TEST ROUTINE
                ; WRITTEN BY DAN FISCHLER
                ;
                ; INITIALIZE VB3 TO 80 X 16
                ; NON-INTERLACED FORMAT

0100 =          LOC     EQU     0100H   ;START OF PROGRAM
00D0 =          VTAC    EQU     0D0H    ;I/O ADDRESS OF THE CRT CONTROLLER
00E0 =          KSTAT   EQU     0E0H    ;BOARD ENABLE
00E1 =          KDATA   EQU     KSTAT+1 ;BOARD DISABLE

0100                    ORG     LOC

0100 D3E0       BEGIN:  OUT     KSTAT   ;ENABLE BOARD

0102 D3DE               OUT     VTAC+14 ;RESET VTAC
0104 D3DA               OUT     VTAC+10

0106 3E70               MVI     A,70H
0108 D3D0               OUT     VTAC

010A 3E53               MVI     A,53H
010C D3D1               OUT     VTAC+1

010E 3E65               MVI     A,65H
0110 D3D2               OUT     VTAC+2

0112 3E0F               MVI     A,0FH
0114 D3D3               OUT     VTAC+3

0116 3E03               MVI     A,03H
0118 D3D4               OUT     VTAC+4

011A 3E26               MVI     A,26H
011C D3D5               OUT     VTAC+5

011E 3E0F               MVI     A,0FH
0120 D3D6               OUT     VTAC+6

0122 3E00               MVI     A,00H
0124 D3DC               OUT     VTAC+12

0126 3E0F               MVI     A,0FH
0128 D3DE               OUT     VTAC+14

012A D3E1               OUT     KDATA   ;DISABLE BOARD

012C C32C01     LOOP:   JMP     LOOP    ;END PROGRAM

012F                    END
```

HEX DUMP

```
0100:  D3  E0  D3  DE  D3  DA  3E  70  D3  D0  3E  53  D3  D1  3E  65
0110:  D3  D2  3E  0F  D3  D3  3E  03  D3  D4  3E  26  D3  D5  3E  0F
0120:  D3  D6  3E  00  D3  DC  3E  0F  D3  DE  D3  E1  C3  2C  01
```

7.0 CIRCUIT DESCRIPTION

7.1 DISPLAY MEMORY ADDRESSING

The display memory portion of the VB3 board appears to the host CPU as up to 8K of random access memory. For the host to access this memory, SELECT (U25 pin 9) must be driven low. This is accomplished by a memory read or write to the 8K block of memory addressed by switch S4. Once SELECT is active, then either VCS0 or VCS1 will go low, depending on A12. VCS0 will enable the first 4K block of memory and VCS1 will enable the second 4K block. SELECT will also allow the host address lines A0 thru A11 onto the internal address bus. SELECT also enables either VCS2 or VCS3 to allow the S-100 data bus onto either the low or high 8 bits of the internal 16 bit data bus.

7.2 CRT CONTROLLER ADDRESSING

The CRT controller is accessed when VTAC is active. This is done by matching A7 thru A4 with the switch setting of switch S4 and performing some kind of I/O request. VTAC is used to select the controller, as well as to enable the data bus drivers. A3 thru A0 are the "Register Selects/Command Code" of the controller. (NOTE: There is no protection against writing to a controller input port. Doing so will cause both the CRT controller and the data bus drivers to be driving the D0 thru D7 data lines at the same time.)

7.3 KEYBOARD, STATUS, AND BOARD ENABLE ADDRESSING

Two input and two output ports are required for the keyboard input, board status, and the board enable switch. One of these functions is enabled when the "B" input of the 1-of-4 selector (74LS139) U18 goes high.

In the case of an "OUT" instruction, the "Y2" output (U18 pin 6) will go low, latching the output of U24 pin 11. If the value that is latched is a 1, then the board is enabled; if a 0, it is disabled. The board enable flag is available on Bit 1 of the board status.

When an "INP" instruction occurs, U24 pin 11 determines what type of input will occur. If it is a 0, then it is a request for keyboard data; if a 1, it is a request for board status. The keyboard data port uses an 8212 strapped as an input port. The data will be latched into the 8212 on the falling edge of the strobe input. The interrupt request line of the 8212 will go low to indicate the data is available. This line can be statused by looking at bit 0 of the board status.

A request for board status will enable the tri-state drivers on U17 to put the keyboard data available bit, the board enable bit and the retrace bit onto the data bus. The retrace bit represents the blanking output of the CRT controller. It of course is used to blank the screen during video retracing, and can also be used by the software to prevent accessing the video memory while the raster is in an active video area.

7.4 DOT AND CHARACTER CLOCK GENERATION

The dot clock (DCLK) is derived directly from the crystal. This is currently set at 16 MHz but may go as high as 20 MHz for special purposes (For example, line lengths of other than 80 columns.). The dot clock is divided down by a 74LS163 (U7) to generate the character clock (DCC or LD or $\overline{LD}$). The divisor of the dot clock is determined by switch S2. Since the 74LS163 is an up-counter, the number of dots-per-character is actually difference in the number of counts between the number loaded and 16. Therefore, to set a character size of 9 dots, 7 (16 minus 9) must be loaded into the divider with switch S2 (see Section 4.2).

7.5 INTERNAL ADDRESS BUS AND THE MAPPER PROM'S

The internal address bus, VA0 through VA11, is normally driven by the CRT controller and used to address the 4K by 16 bit memory. The column counters of the controller, H0 through H3, go directly to VA0 thru VA3. The counters, H4 thru H6, DR0 thru DR4, and H7/DR5, are first sent through the mapper PROM's U3 and U4 to become VA4 through VA11. The purpose of the mapper PROM's is to eliminate the gaps that would normally occur between lines. These gaps occur naturally because the CRT controller uses row/column addressing instead of binary addressing. The exact function of the mapper PROM's is:

$$VA = DR * X + H$$

where VA is the number formed by the bits VA11 through VA4
DR is the number formed by the bits DR5 through DR0
H is the number formed by the bits H7 through H4
and X is the number of columns to be displayed, divided by 16 and rounded up to the next integer

Note that H7 and DR5 are on the same physical pin of the CRT controller. The pin H7/DR5 is H7 when there are more than 128 columns, and DR5 when there are more than 32 rows. Because of this, programming ROM's for 132 columns gets a bit tricky. SSM has available ROM's for non-standard screen sizes, including 132 columns, as an option.

7.6 INTERNAL DATA BUS

The internal data bus is a 16-bit bus and runs between the memory, the S-100 bus drivers and the 8002 Video Generator. VD7 through VD0 carry the 8-bit code of the character to be displayed. This code may be further modified by VD15 through VD8, the attribute byte. The host CPU may access the lower half of the data bus by addressing the first (bottom) 4K of the display memory. The upper half of the data bu is on the second (top) 4K of the display memory.

7.7 CHARACTER AND ATTRIBUTE GENERATION

Sixteen bits of data are used to control the visual appearance of each character on the screen. The 8 least significant bits are used to address the character ROM's and the 8 most significant bits are

attributes that may be applied to the character. The mode bits (bits 8 and 9 of the 16-bits; bits D0 and D1 of the most significant bits) are decoded to detect the external mode. When external mode occurs, the tri-state drivers (U28 and part of U21) are turned off and the character generator EPROM (2716 or 2732) is turned on.

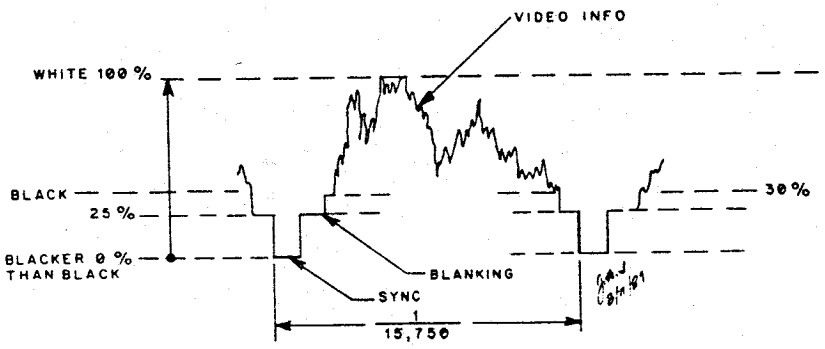
7.8 SCREEN BLANKING DURING MEMORY ACCESS

During host access of the video memory, the video is blanked to prevent white "snow" on the screen. This is done by OR'ing the select signal into the blanking signal. Because of memory delays, the effects of a host select are not felt until one character time after the select actually occurs. This is why the select signal is shifted by the character clock before it is used to blank the screen. The actual blanking time will vary from 0 to 2 character times, depending on the speed of the host CPU.

7.9 COMPOSITE VIDEO GENERATION

Composite video is created by combining VSYNC, HSYNC, VIDEO and intensity control signals together at the base of transistor Q1. Transistor Q1 is connected as an emitter follower, which has no voltage gain but does have current gain to provide a low impedance drive to R5 and R6. Q1, R5, and R6 provide a 50 to 75 ohm output impedance for driving standard 75 ohm video coaxial cable.

The composite video signal is an amplitude-modulated signal that controls the vertical and horizontal sync pulses, the vertical and horizontal blanking pulses, and the video information signal. All are at different levels within an approximately 1 volt (peak-to-peak) range. "Blacker-than-black" is a range from 0 volts up to approximately 25% of the amplitude of the composite video signal. The vertical and horizontal sync pulses are within the "blacker than black" range, and are detected and seperated by the TV monitor to synchronize its internal oscillators. About 5% above "blacker than black" range is the true "black level" range for video picture information. Within the 5% "black level" range is the horizontal and vertical blanking level, which is a zone that the monitor uses to return the electron beam across the picture tube unseen, to get ready to start another scan line. (Some video products provide a special blanking level. For example, the SSM VB2 provides special blanking levels, but the VB3 does not).



The 100% point of the possible video level is maximum white for the composite video signal.

Horizontal and vertical sync are combined through a gate (U6, pins 9 and 10) which in turn drives two open-collector inverters (U10, pins 1 and 13). If either sync signal goes high, then U10 (pins 2 and 12) drives the base of Q1 to zero volts for a "blacker than black" level.

Video information is passed through an inverter (U6, pin 12) to obtain the correct polarity for black and white control on U10 pin 3. If VIDEO equals a logic one (white), then U10 pin 4 is high (off) and Q1 generates at its output the maximum level for white. If VIDEO equals a logic zero (black), then U10 pin 4 is low (on) and Q1 receives at its input (base pin) a voltage divided by R7 and R10, producing a black level.

Reduced intensity (gray) is controlled by U10 pin 6 and R11. If the video is white and gray is requested, then U10 pin 4 is off and U10 pin 6 is on (low). Q1 receives at its input (base pin) a voltage divided by R7, with R10 added to R11. This level is somewhere between black and white, and the level of gray is controlled by the value of R11.

7.10 EXTERNAL SYNC

The VB3 can be synchronized to an external video standard for image overlay with another video signal. The horizontal and vertical sync inputs (see Section 4.1) allow an external timing pulse (at TTL levels) to be brought into the VB3. The circuitry used is per *SMC's Application Note for the 5037 (VTAC)

The horizontal sync circuit controls the crystal oscillator circuit. At the end of a horizontal scan line, HSYNC (from U1) gets strobed into U8 as a logic one. U8 pin 8 goes low (0) and when LD goes low, then U9 pin 10 goes high (1). U9 pin 10 controls the crystal oscillator and must be low to allow normal operation. When U9 pin 10 is high, the oscillator stops; when low, the oscillator runs. When stopped, the oscillator can be re-started by providing a negative-going pulse to U8 pin 13. The external sync pulse is sent through U6 pin 2 to U6 pin 3 (which controls the polarity) to U8 pin 13, which starts the oscillator, and begins the next horizontal scan line.

For good operation:

(1) Set the scan rate of the 5037 slightly higher than the external horizontal sync standard.

16000000/(DOTS x (HCOUNT + 3)) => External Horizontal Rate (Hz)

---

* Standard Microsystems Corp., 35 Marcus Blvd., Hauppauge, NY 11787

(2) Set HSP (see Section 5.1, Register 1) to 1.

Example: If the horizontal rate you wish to synchronize to is 15,734.26 Hz, then the horizontal rate of the VB3 must be greater. Assume 9 dots for character width.

16000000/(9 x (HCOUNT + 3)) => 15734.26

(HCOUNT + 3) <= (16000000/(15734.26 x 9))

HCOUNT <= (16000000/141608.3)-3

HCOUNT <= 113 - 3

HCOUNT = 110

The vertical sync circuit controls the dot counter (U7). At the end of one field (typically 1/60 sec.), VSYNC (from U1) gets strobed into U8 as a logic 1. U8 pin 6 goes low (0). When LD goes low, then U20 pin 4 goes low. U20 pin 4 controls the dot counter and must be high to allow normal operation. When U20 pin 4 is low, the counter is stopped; when high, it is counting. The counter, when stopped, can be re-started by providing a negative-going pulse to U8 pin 1. The external sync pulse is sent through U6 pin 5 to U6 pin 6 (which controls the polarity) to U8 pin 1. The counter is started by the external sync pulse, which begins the next vertical field.

For good operation:

Set the vertical rate of the 5037 slightly higher than the external vertical sync standard.

16000000/Dots x HCOUNT x SCANS => External Vertical Rate (Hz)

Example: If the vertical rate you wish to synchronize to is 60 Hz, then the vertical rate of the VB3 must be greater. This can be done by changing the horizontal rate and leaving the number of scan lines the same. (The previous equation assumes non-interlaced video, so divide the "scan" number by two if interlaced.)

Assume: Dots = 9, Scans = 525/2, HCOUNT = 110

16000000/(9 x 110 x 262.5) => 60 Hz

16000000/259875 => 60 Hz

61.56 => 60 Hz

8.0 TROUBLESHOOTING HINTS

This section assumes the user has some basic knowledge of logic circuits and has read the Theory of Operation section of this manual. It is also assumed that the user has a voltmeter and a logic probe or an oscilliscope.

[ ] Verify that all IC's are in the correct sockets and that none of the pins are bent under the package or out from the socket. Also, check to see that the IC is installed with the proper orientation.

[ ] Verify that each regulator (X1, X2, X3, and X4) is outputting +5v (+/- 0.2v). Verify that Q2 is outputting +12v (+/- 0.2v).

[ ] Verify that all jumper options and switch settings are set correctly.

[ ] Inspect the back side of the board for any solder bridges or shorts. Run a small sharp knife blade between traces that may appear suspicious. A magnifying glass is recommended for this inspection.

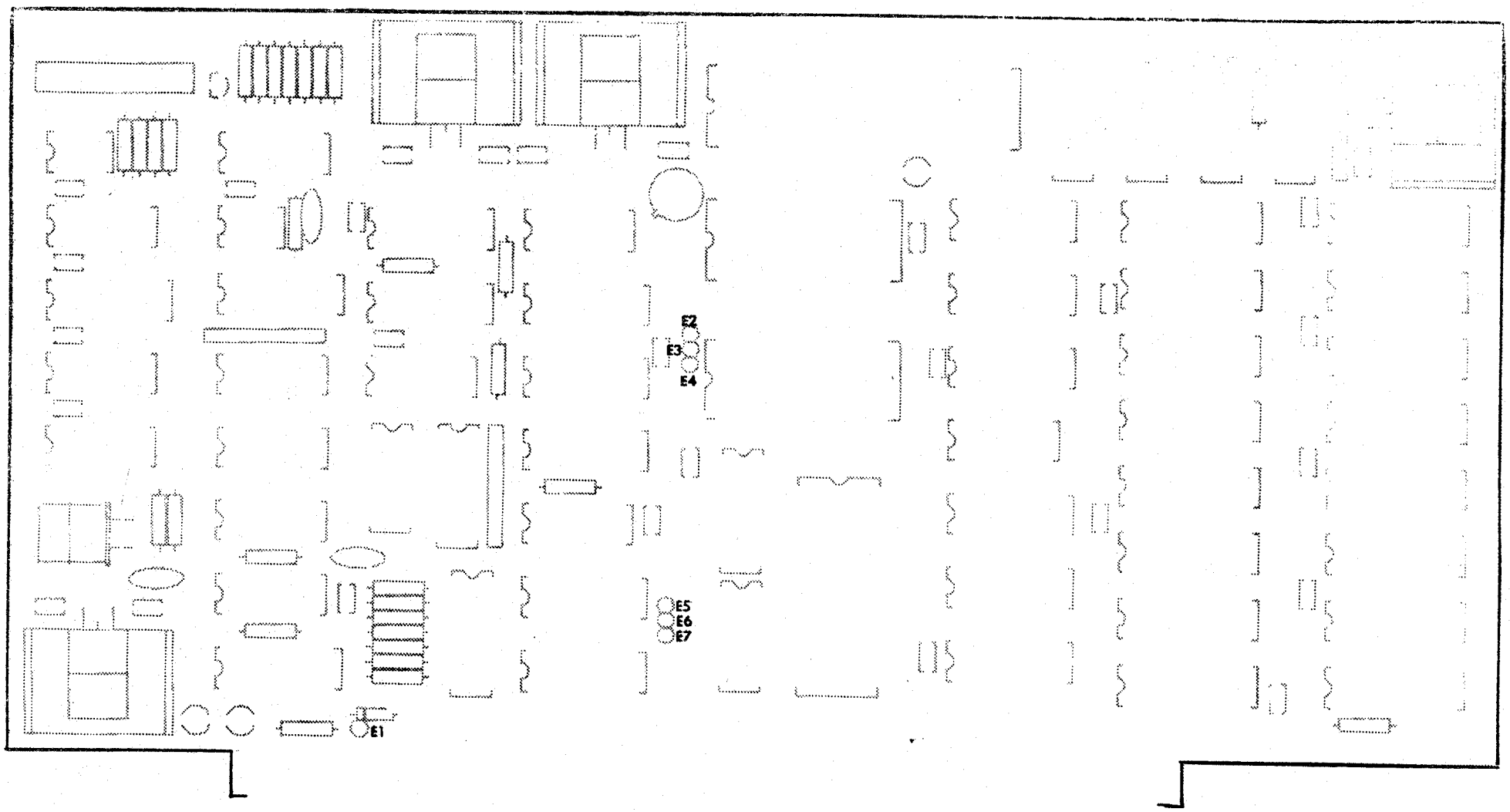
9.0 WARRANTY

SSM Microomputer Products, Inc. warrants its products to be free from defects in materials and/or workmanship for a period of 90 days for kits and one (1) year for factory assembled boards. In the event of malfunction or other indication of failure attributable directly to faulty workmanship and/or material, then, upon return of the product (postage paid) to SSM at 2190 Paragon Drive, San Jose, California 95131, "Attention: Warranty Claims Department", SSM will, at its option, repair or replace the defective part or parts to restore said product to proper operating condition. All such repairs and/or replacements shall be rendered by SSM without charge for parts or labor when the product is returned within the specified period of the date of purchase. This warranty applies only to the original purchaser.

This warranty will not cover the failure of SSM products which at the discretion of SSM, shall have resulted from accident, abuse, negligence, alteration, or misapplication of the product. While every effort has been made to provide clear and accurate technical information on the application of SSM products, SSM assumes no liability in any events which may arise from the use of said technical information.

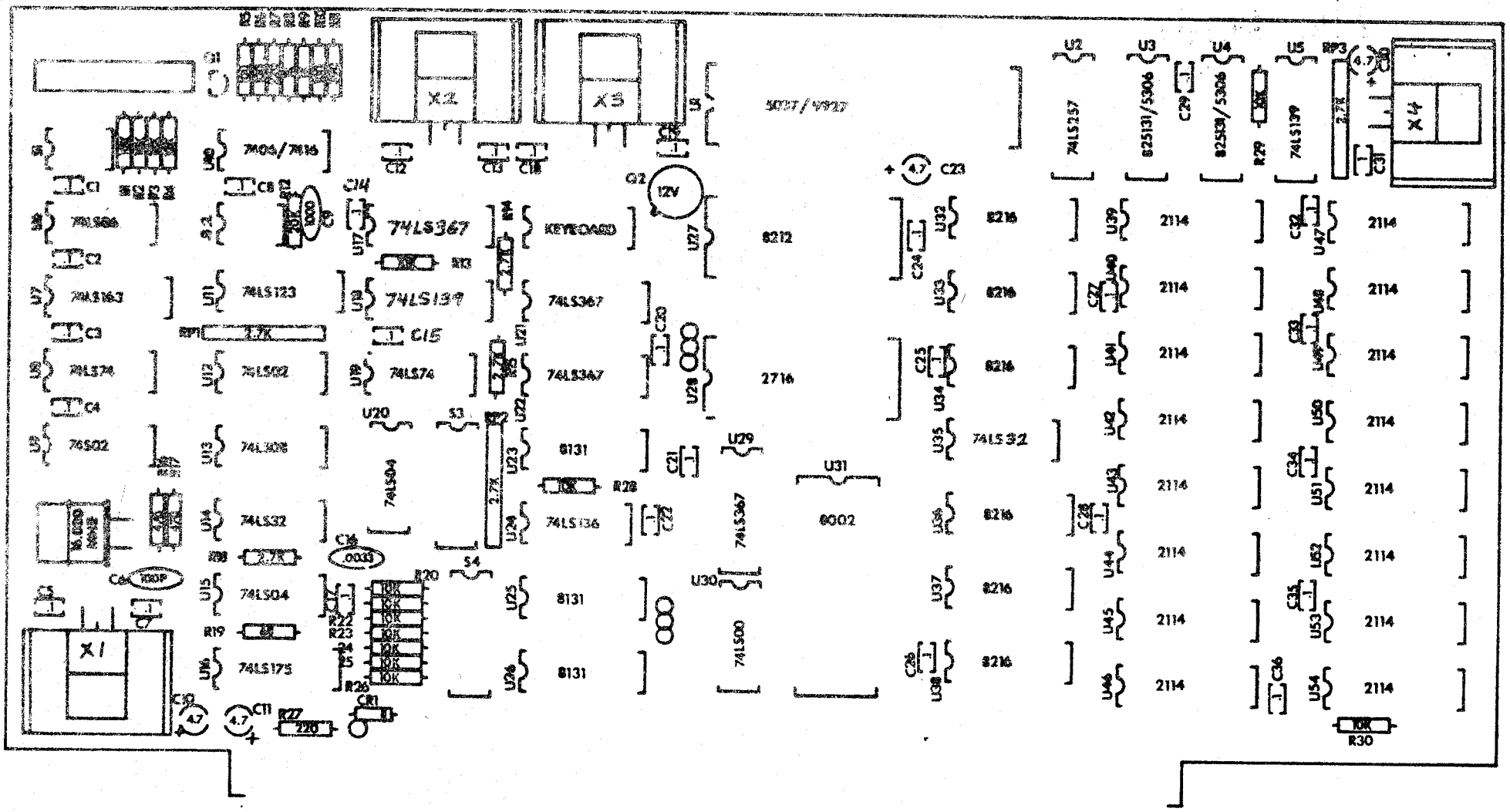
This warranty is in lieu of all other warranties, expressed or implied, including warranties of mercantability and fitness for use. In no event will SSM be liable for incidental and consequential damages arising from or in any way connected with the use of its products. Some states do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

IMPORTANT: Proof of purchase is necessary for products returned for repair under warranty. Before returning any product please call our Customer Service Department for a return authorization number.



JUMPER DRAWING

VB3

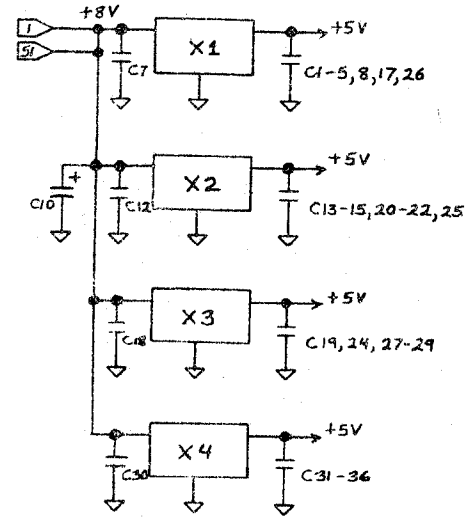


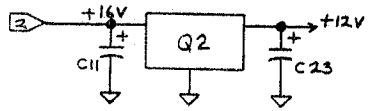
ASSEMBLY DRAWING

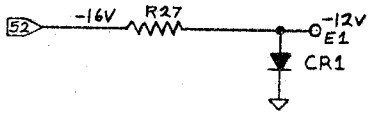
VB3

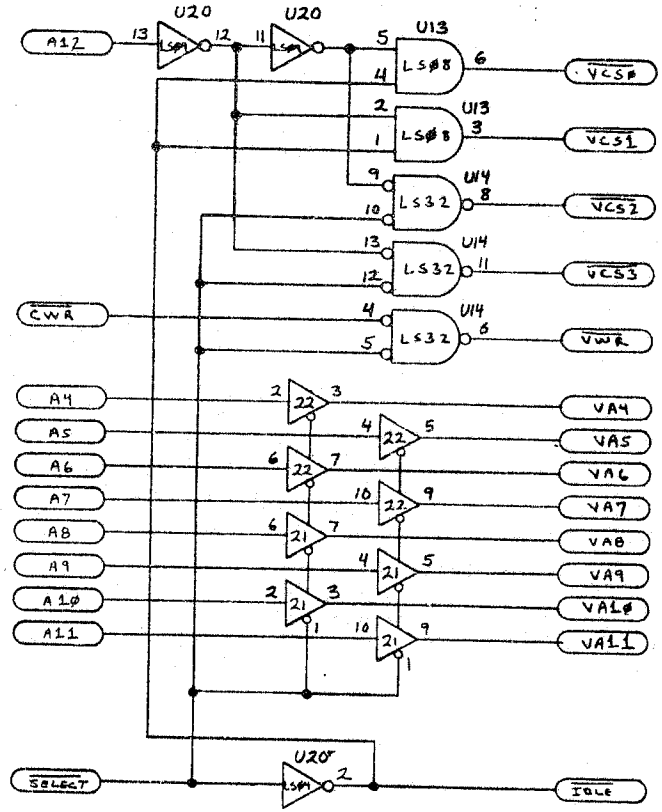
APPENDIX

ASSEMBLY DRAWING
JUMPER DRAWING
SCHEMATICS
VB3 PARTS LIST

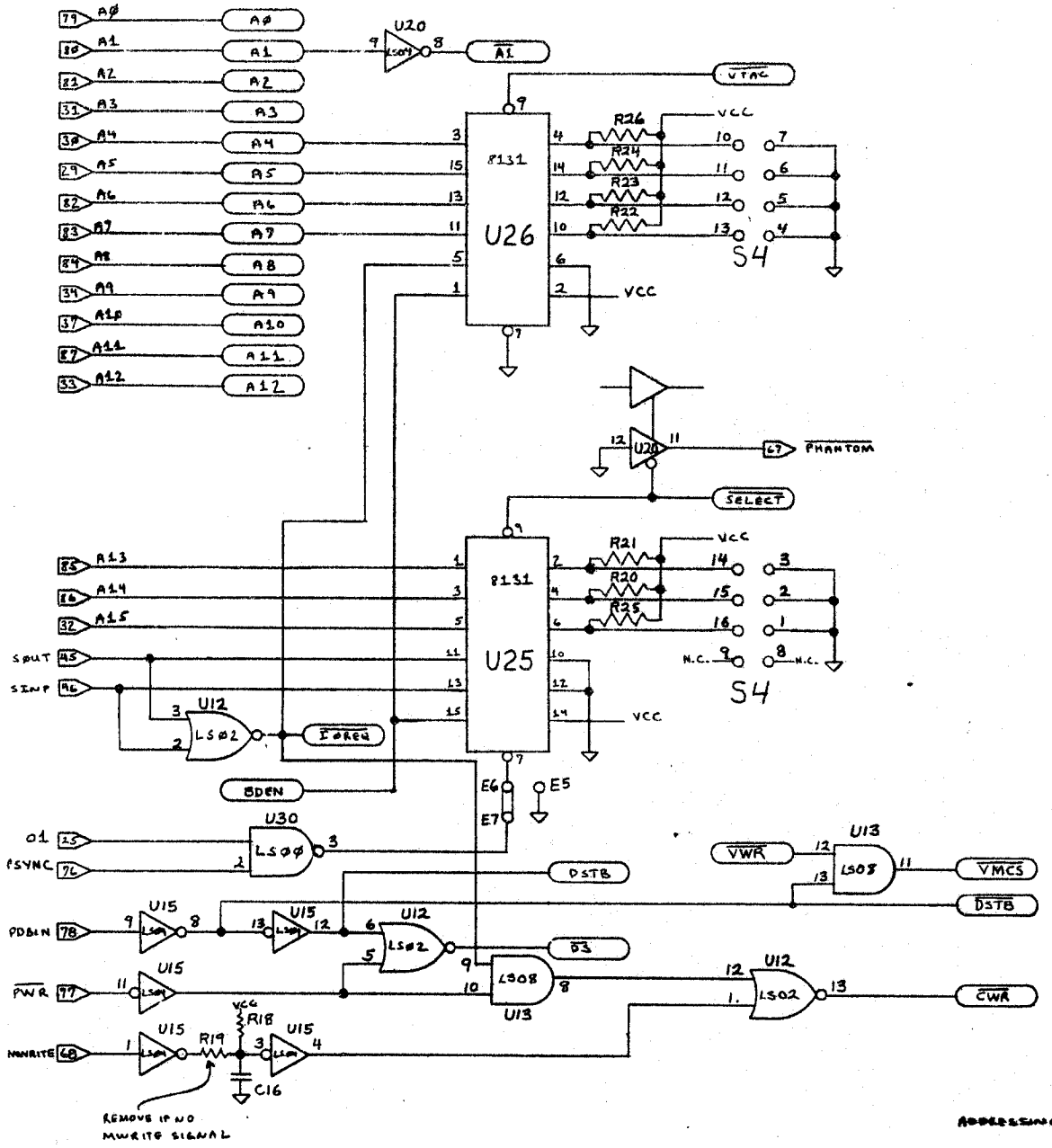


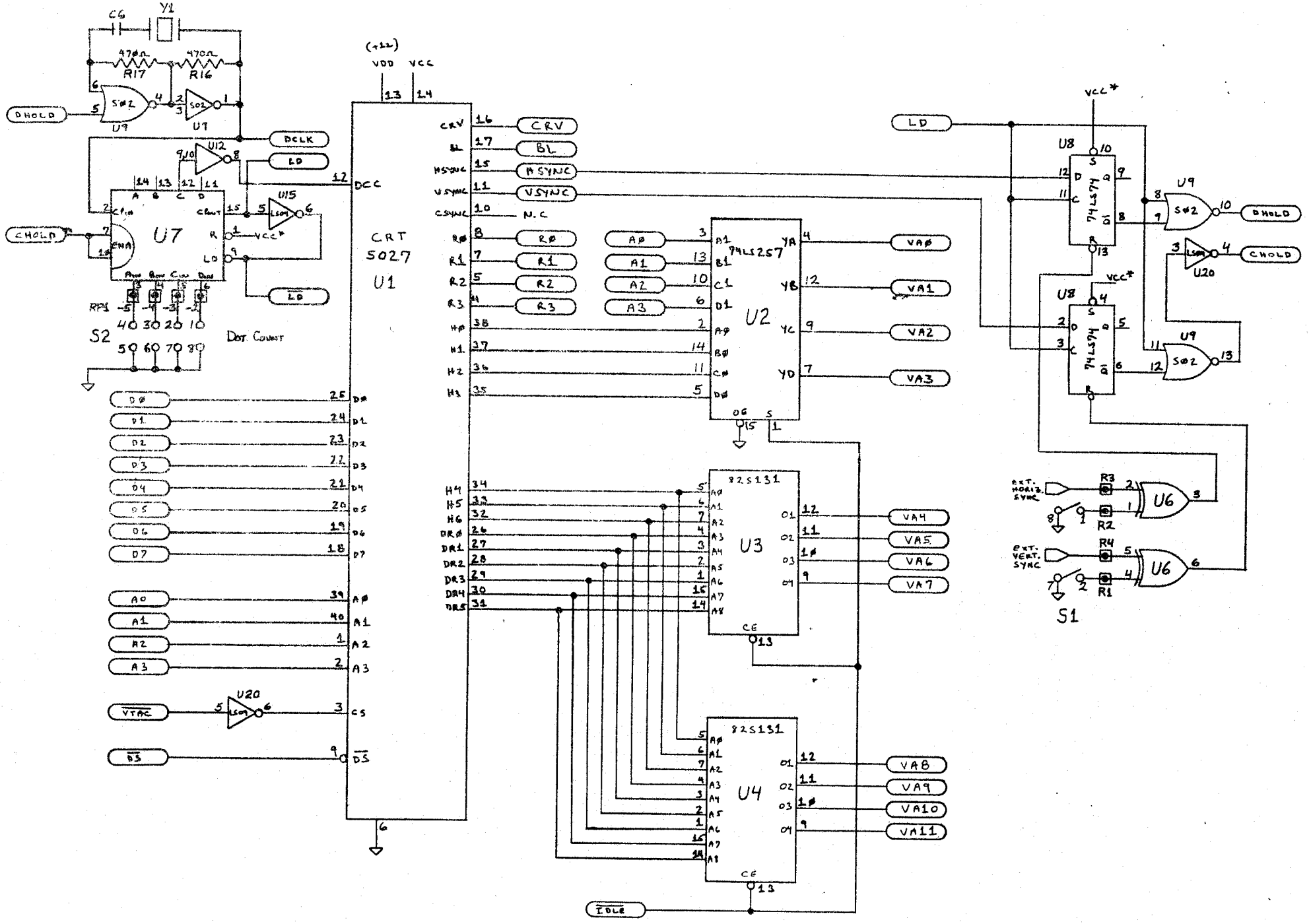




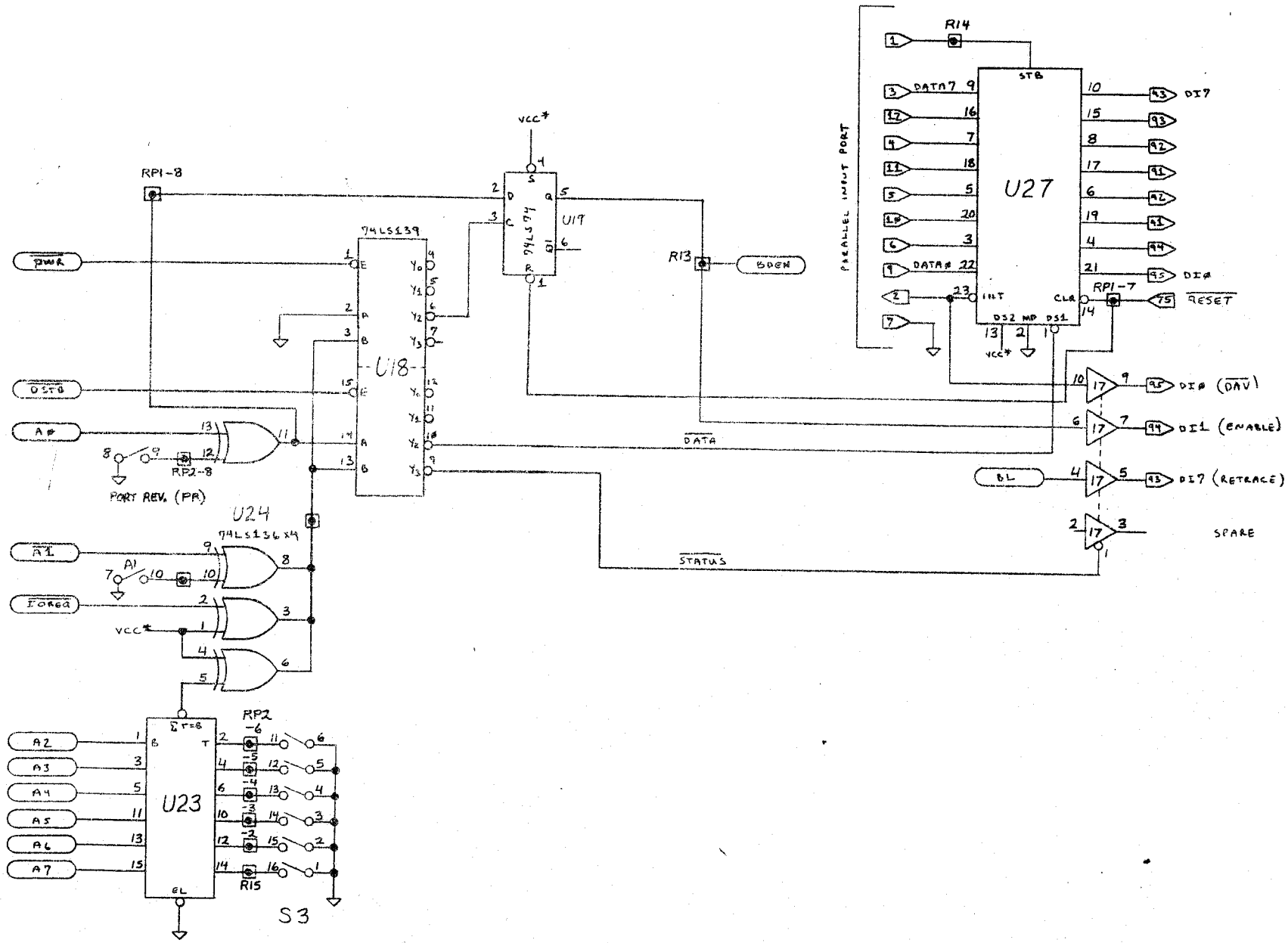


REGULATORS & VIDEO CONTROLS

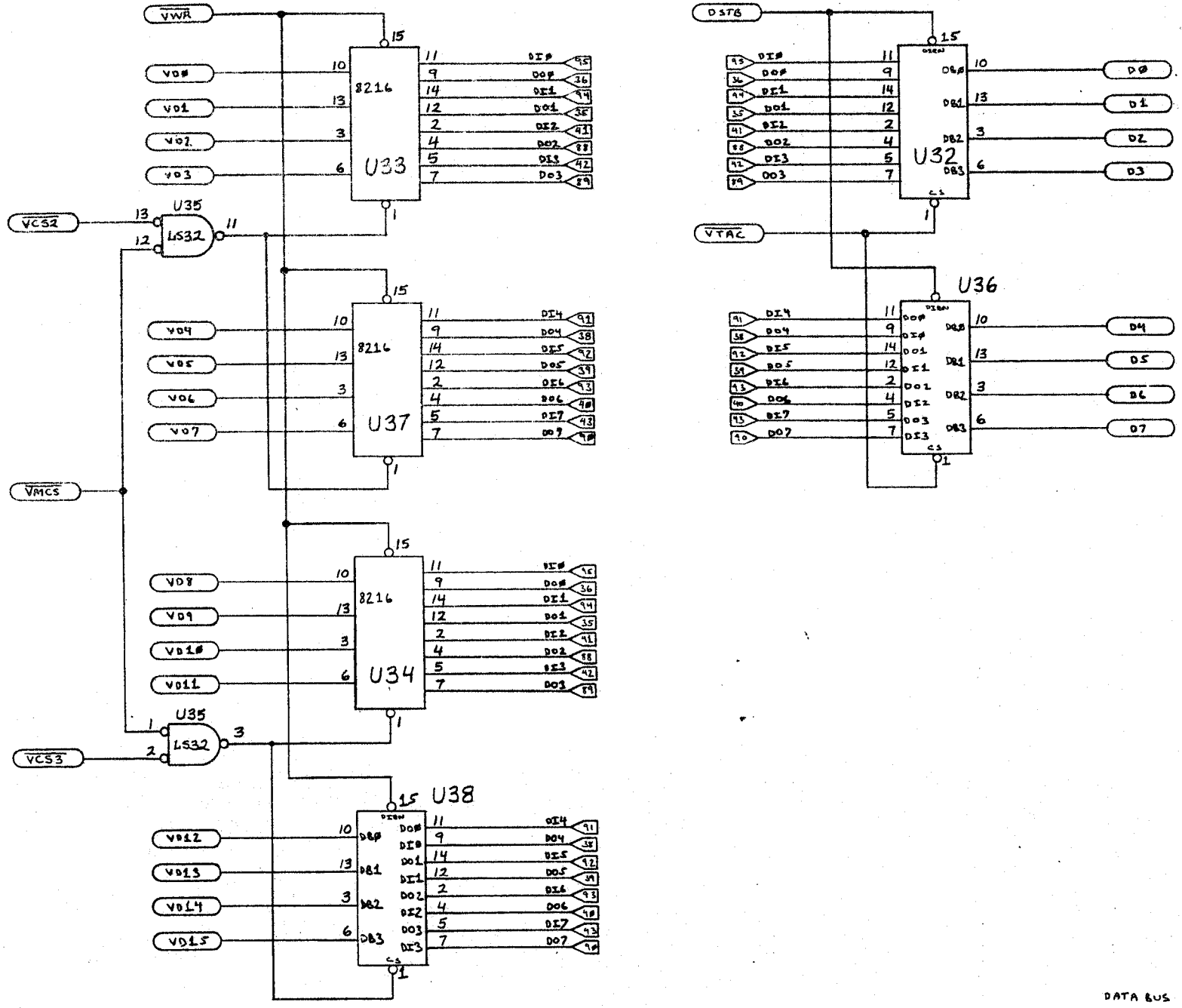

A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
U20
LS04
8131
U26
R26
R24
R23
R22
VCC
S4
VTAC
PHANTOM
SELECT
A13
A14
A15
SOUT
SINP
U25
R21
R20
R25
N.C.
U12
LS02
IORQ
BDEN
E6
E7
E5
U30
LS00
01
PSYNC
DSTB
U15
LS04
PDBIN
PWR
MWRITE
R18
R19
C16
REMOVE IF NO MWRITE SIGNAL
U13
LS08
VWR
VMCS
LS32
CWR
D3
ADDRESSING AND CONTROLS
PAGE 1 OF 7
© SSM 1980



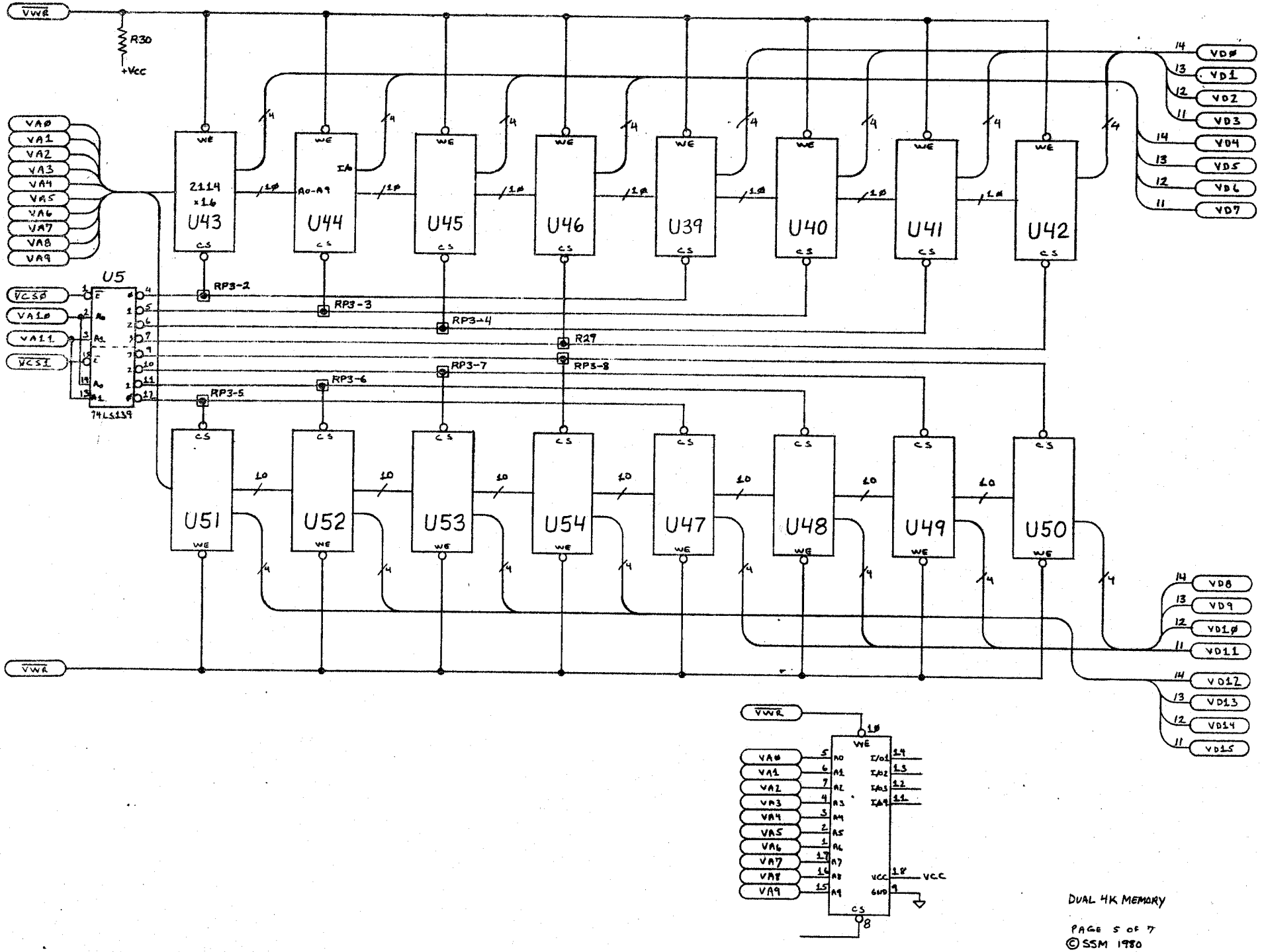
VIDEO TIMING

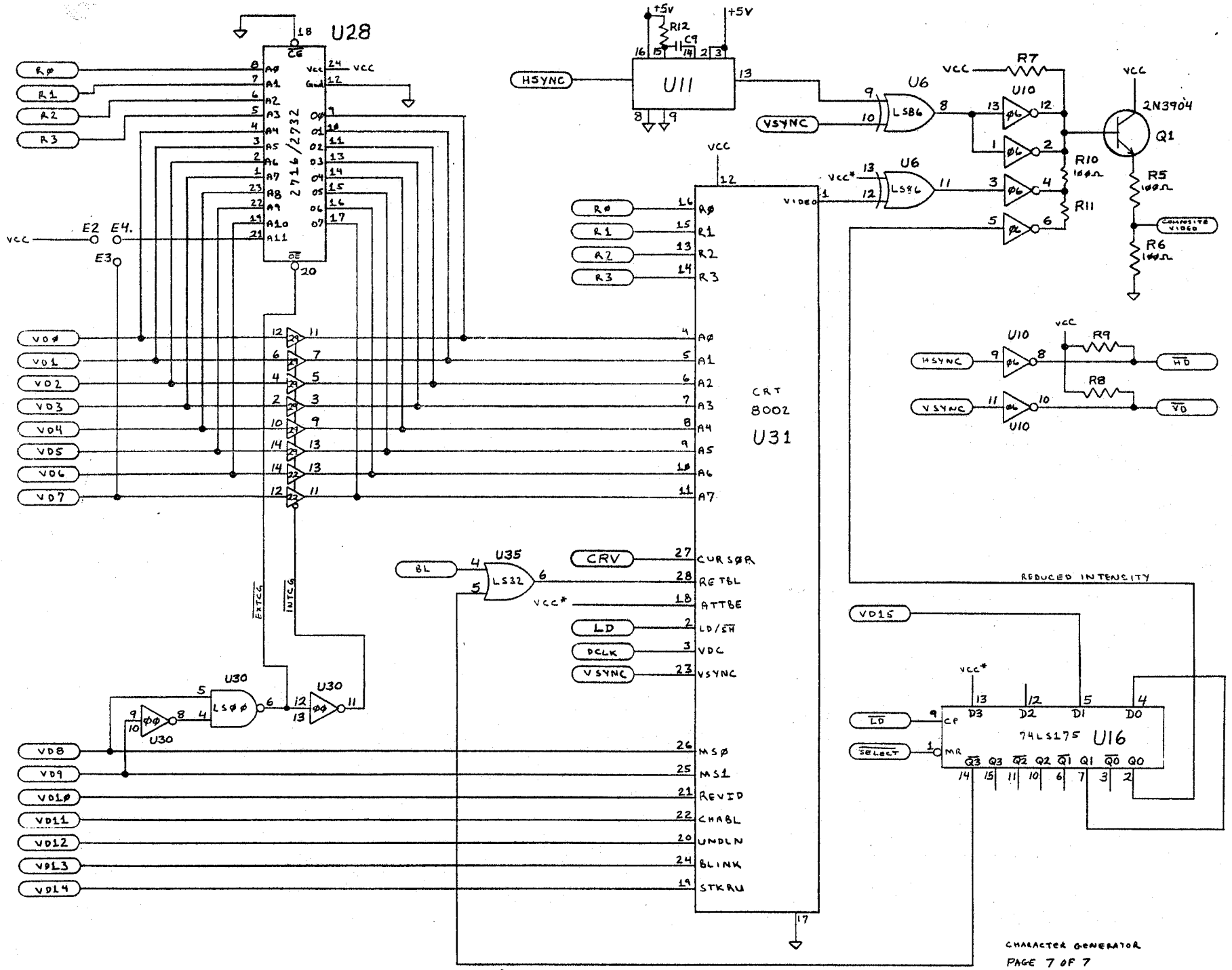


KEYBOARD & BANK-SELECT



DATA BUS INTERFACE
PAGE 6 OF 7


VWR
R30
+Vcc
VA0
VA1
VA2
VA3
VA4
VA5
VA6
VA7
VA8
VA9
2114
x4
U43
U44
U45
U46
U39
U40
U41
U42
WE
CS
A0-A9
I/O
VD0
VD1
VD2
VD3
VD4
VD5
VD6
VD7
U5
VCS0
VA10
VA11
VCS1
74LS139
RP3-2
RP3-3
RP3-4
R27
RP3-8
RP3-7
RP3-6
RP3-5
U51
U52
U53
U54
U47
U48
U49
U50
VD8
VD9
VD10
VD11
VD12
VD13
VD14
VD15
VCC
GND
DUAL 4K MEMORY
PAGE 5 OF 7
©SSM 1980


U28
2716/2732
U11
U6
LS86
U10
R7
R12
C9
+5V
VCC
2N3904
Q1
R10
R11
R5
R6
COMPOSITE VIDEO
HSYNC
VSYNC
U31
CRT
8002
U35
LS32
U30
LS00
R9
R8
REDUCED INTENSITY
U16
74LS175
CHARACTER GENERATOR
PAGE 7 OF 7
© SSM 1980

PARTS LIST

CHIP PACK

1-- U2	74LS257	quad data selector/multiplexer
2-- U5, 18	74LS139	dual 2-to-4 line decoder
1-- U6	74LS86	quad 2-input exclusive-OR
1-- U7	74LS163	synchronous 4-bit binary counter
2-- U8, 19	74LS74	dual D-type flip flop
1-- U9	74S02	quad 2-input NOR gate
1-- U10	7416	hex inverting buffer w/OC outputs
1-- U11	74123	dual monostable multivibrator
1-- U12	74LS02	quad 2-input NOR gate
1-- U13	74LS08	quad 2-input AND gate
2-- U14, 35	74LS32	quad 2-input OR gate
2-- U15, 20	74LS04	hex inverter
1-- U16	74LS175	quad D-type flip flop
4-- U17, 21, 22, 29	74LS367	hex bus driver
1-- U24	74LS136	quad exclusive OR gate
3-- U23, 25, 26	DM8131	6-bit comparator
1-- U27	8212	8-bit input/output port
1-- U30	74LS00	quad 2-input NAND gate
6-- U32-34, 36-38	8216	4-bit Bi-directional driver

MEMORY PACK

1-- U1	SMC 5037	CRT Video Timer &Controller
1-- U3	74S571	256 x 4-bit PROM (80-L)
1-- U4	74S571	256 x 4-bit PROM (80-H)
1-- U28	2716	2048 x 8-bit EPROM (6 x 7)
1-- U31	SMC 8002	CRT Attributes Controller
8-- U39, 40, 43, 44, 47, 48, 51, 52	2114L-2	4096 x 1-bit static RAM

CAPACITOR PACK

29-- C1-5, 7, 8, 12-15, 17-22, 24-29, 31-36	.1 uf monolithic filter capacitor
1-- C6	100 pf disk radial
1-- C9	1000 pf/.001 uf disc radial
4-- C10, 11, 23, 30	4.7 uf 25v dipped tantalum radial
1-- C16	.0033 uf disc radial

RESISTOR PACK

15-- R1-4, 13, 20-26, 28-30	10K ohm 1/4W 5%	(brown, black, orange)
3-- R5, 6, 10	100 ohm 1/4W 5%	(brown, black, brown)
2-- R7, 27	220 ohm 1/4W 5%	(red, red, brown)
2-- R8, 9	1K ohm 1/4W 5%	(brown, black, red)
1-- R11	390 ohm 1/4W 5%	(orange, white, brown)
1-- R12	20K ohm 1/4W 5%	(red, black, orange)
3-- R14, 15, 18	2.7K ohm 1/4W 5%	(red, violet, red)
2-- R16, 17	470 ohm 1/4W 5%	(yellow, violet, brown)
1-- R19	68 ohm 1/4W 5%	(blue, grey, black)
3-- RP1-3	2.7K ohm SIP resistor network	

HARDWARE PACK

3-- X1-3	Heatsink
1-- X4	Special Heatsink
4--	Sets of #6 hardware
1--	20 pin right-angle connector
1--	20 pin shell connector
10--	Connector pins (crimp type)
8--	wire-wrap pins

REGULATOR PACK

1-- Q1	2N3904/2N2222 transistor
1-- Q2	78L12AC regulator
4-- X1-4	7805 +5v volt regulator
1-- Y1	16 MHz crystal
1-- CR1	1N5242/1N4742 12v zener diode

SOCKET PACK

6--	18 pin sockets
4--	14 pin sockets
1--	16 pin sockets
2--	24 pin sockets
1--	28 pin sockets
1--	40 pin sockets
2-- S1, 2	4 position DIP switch
2-- S3, 4	8 position DIP switch

MISCELLANEOUS

10--	18 pin sockets
10--	14 pin sockets
20--	16 pin sockets
1--	VB3 PC Board
1--	VB3 Manual
1--	Warranty Card

NOTE: The VB3 80 x 50 also includes the following parts:

MEMORY PACK

8-- U41, 42, 45, 46, 49, 50, 53, 54	2114L-2	4096 x 1-bit static RAM

SSM VB3 6 x 7 CHARACTER EPROM RETROFIT

Instruction Sheet

SSM MICROCOMPUTER PRODUCTS, INC.
2190 Paragon Drive
San Jose, California 95131

Telephone: (408) 946-7400
TWX: 910-338-2077
Telex: 171171
DDD: (408) 946-3644 (110 Baud)

Part No. MN0027

1.0 INTRODUCTION

The SSM VB3 6x7 CHARACTER EPROM RETROFIT provides a new character set of reduced height for the VB3. The standard SMC 8002 character set is composed of 7x9 pixels in a 7x11 character field, which provides for descenders of 2 pixels. This optional EPROM provides a character set of 6x7 pixels in a 6x9 character field (also providing for descenders of 2 pixels).

To place this in perspective, the optional EPROM provides a 22 percent reduction in the number of required raster lines to display a character row. This will provide the following screen formats:

P4 or P31 Phosphor, Non-interlace, 262 raster lines
Standard character set (SMC 8002) . . up to 20 lines of text
Optional EPROM . . . . . up to 25 lines of text

P39 Phosphor, Interlaced, 525 raster lines
Standard character set (SMC 8002) . . up to 38 lines of text
Optional EPROM . . . . . up to 50 lines of text

2.0 INSTALLATION/SET-UP

2.1 IC INSTALLATION

[ ] Referring to the Assembly Drawing in the APPENDIX, install the 2716 EPROM (marked "VB3 6x7") in IC position U28. Be **certain that pin 1 of the IC is to the LEFT.** It is sometimes helpful to bend the leads of the IC **slightly** inward by placing the circuit on its side and applying firm pressure. This assures that the leads will be straight and makes it easier to install the device in a socket.

Be sure that all pins go into the socket and are not under the IC package or bent out from the socket.

2.2 JUMPER SET-UP

[ ] Install a jumper from E2 to E4 on the 3-pin wire-wrap post next to IC position U28. (Connect the top and bottom posts and be sure **NOT** to short to the center post.)

3.0 USAGE

To implement the new character set, the ATTRIBUTE MEMORY (refer to Section 5.8 of the VB3 manual for more detailed information) must have Bit 0 set to a "1" and Bit 1 set to a "0".

For those using the VBIOS or ICRT routines, the following parameters need to be EQUated in the program.

80x24 DISPLAY			VB3 REGISTER	HEX CODE
VTAC	EQU	0D0H	D0 (VTAC)	70
NCOLS	EQU	80	D1	69
NROWS	EQU	24	D2	4D
SKEW	EQU	0	D3	17
SCANF	EQU	262	D4	03
SCANR	EQU	10	D5	0C
HCOUNT	EQU	113	D6	17
CRT5037	EQU	TRUE		
INTERLACED	EQU	FALSE		

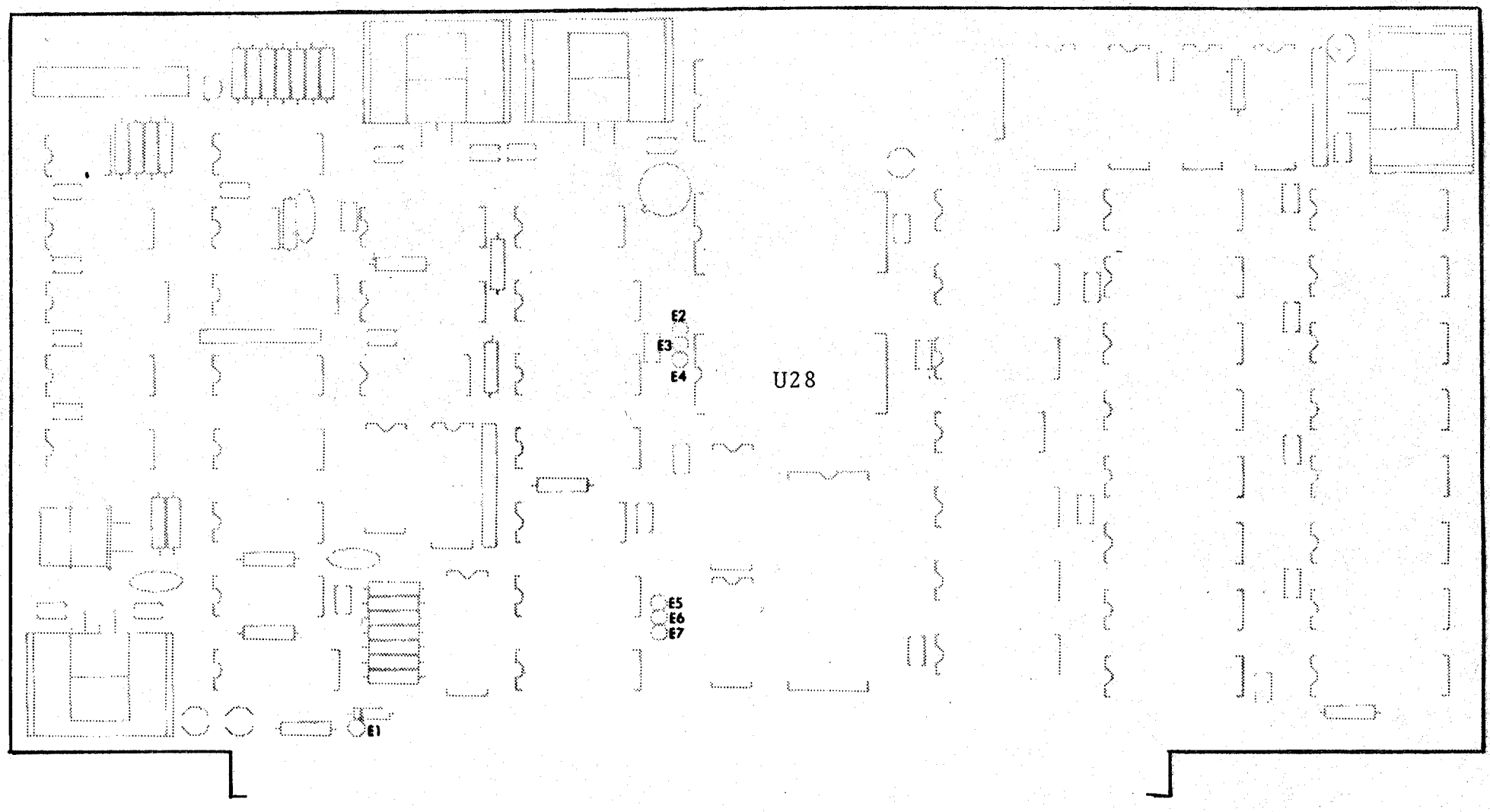
80x50 DISPLAY			VB3 REGISTER	HEX CODE
VTAC	EQU	0D0H	D0	70
NCOLS	EQU	80	D1	E9
NROWS	EQU	50	D2	45
SKEW	EQU	0	D3	31
SCANF	EQU	525	D4	06
SCANR	EQU	10	D5	0C
HCOUNT	EQU	113	D6	31
CRT5037	EQU	TRUE		
INTERLACED	EQU	TRUE		

The attribute called "NORMAL" should be set to 1 instead of 3 to select the new character set.

```
000 44 64 54 4C 44 11 11 11 0E 00 00 00 00 00 00 00
010 38 40 38 04 38 09 0F 09 09 00 00 00 00 00 00 00
020 38 40 38 04 38 09 06 06 09 00 00 00 00 00 00 00
030 78 40 70 40 78 09 06 06 09 00 00 00 00 00 00 00
040 78 40 70 40 78 1F 04 04 04 00 00 00 00 00 00 00
050 78 40 70 40 78 0F 09 0B 0F 00 00 00 00 00 00 00
060 10 28 44 7C 45 0A 0C 0A 09 00 00 00 00 00 00 00
070 00 08 1C 22 22 22 7F 08 00 00 00 00 00 00 00 00
080 70 48 70 48 7E 10 0E 01 0E 00 00 00 00 00 00 00
090 48 48 78 48 48 1F 04 04 04 00 00 00 00 00 00 00
0A0 40 40 40 40 78 0F 08 0E 08 00 00 00 00 00 00 00
0B0 44 44 44 28 10 1F 04 04 04 00 00 00 00 00 00 00
0C0 78 40 70 40 4F 08 0E 08 08 00 00 00 00 00 00 00
0D0 38 40 40 38 0E 09 0E 0A 09 00 00 00 00 00 00 00
0E0 38 40 38 04 38 06 09 09 06 00 00 00 00 00 00 00
0F0 38 40 38 04 38 07 02 02 07 00 00 00 00 00 00 00
100 78 24 24 24 78 08 08 08 0F 00 00 00 00 00 00 00
110 78 44 44 78 0C 04 04 04 0E 00 00 00 00 00 00 00
120 78 44 44 78 1E 01 0E 10 1F 00 00 00 00 00 00 00
130 78 44 44 78 1E 01 0E 01 1E 00 00 00 00 00 00 00
140 78 44 44 78 0A 0A 0F 02 02 00 00 00 00 00 00 00
150 48 68 58 48 49 0A 0C 0A 09 00 00 00 00 00 00 00
160 38 40 38 04 38 11 0A 04 04 00 00 00 00 00 00 00
170 78 40 70 40 7E 09 0E 09 0E 00 00 00 00 00 00 00
180 38 40 40 40 38 09 0D 0B 09 00 00 00 00 00 00 00
190 78 40 70 40 78 11 1B 15 11 00 00 00 00 00 00 00
1A0 38 40 38 04 3E 09 0E 09 0E 00 00 00 00 00 00 00
1B0 78 40 70 40 78 07 08 08 07 00 00 00 00 00 00 00
1C0 78 40 70 40 4E 10 0E 01 0E 00 00 00 00 00 00 00
1D0 38 40 4C 44 3E 10 0E 01 0E 00 00 00 00 00 00 00
1E0 78 44 78 50 4E 10 0E 01 0E 00 00 00 00 00 00 00
1F0 44 44 44 44 3E 10 0E 01 0E 00 00 00 00 00 00 00
200 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
210 08 08 08 08 00 08 08 00 00 00 00 00 00 00 00 00
220 12 24 24 00 00 00 00 00 00 00 00 00 00 00 00 00
230 14 14 7F 14 7F 14 14 00 00 00 00 00 00 00 00 00
240 08 3E 48 3E 09 3E 08 00 00 00 00 00 00 00 00 00
250 22 54 24 08 0A 15 12 00 00 00 00 00 00 00 00 00
260 18 24 24 18 25 42 42 3D 00 00 00 00 00 00 00 00
270 04 08 08 00 00 00 00 00 00 00 00 00 00 00 00 00
280 08 10 10 10 10 10 08 00 00 00 00 00 00 00 00 00
290 08 04 04 04 04 04 08 00 00 00 00 00 00 00 00 00
2A0 00 24 18 7E 18 24 00 00 00 00 00 00 00 00 00 00
2B0 00 08 08 7F 08 08 00 00 00 00 00 00 00 00 00 00
2C0 00 00 00 00 00 0C 0C 04 08 00 00 00 00 00 00 00
2D0 00 00 00 3E 00 00 00 00 00 00 00 00 00 00 00 00
2E0 00 00 00 00 00 0C 0C 00 00 00 00 00 00 00 00 00
2F0 01 02 04 08 10 20 40 00 00 00 00 00 00 00 00 00
300 3C 42 46 4A 52 62 3C 00 00 00 00 00 00 00 00 00
310 08 18 08 08 08 08 1C 00 00 00 00 00 00 00 00 00
320 3C 42 02 3C 40 40 7E 00 00 00 00 00 00 00 00 00
330 3C 42 02 1C 02 42 3C 00 00 00 00 00 00 00 00 00
340 44 44 44 44 7E 04 04 00 00 00 00 00 00 00 00 00
350 7E 40 7C 02 02 42 3C 00 00 00 00 00 00 00 00 00
360 3C 40 5C 62 42 42 3C 00 00 00 00 00 00 00 00 00
370 7E 02 04 08 10 20 40 00 00 00 00 00 00 00 00 00
380 3C 42 42 3C 42 42 3C 00 00 00 00 00 00 00 00 00
390 3C 42 42 3E 02 02 3C 00 00 00 00 00 00 00 00 00
3A0 00 00 0C 0C 00 0C 0C 00 00 00 00 00 00 00 00 00
3B0 00 00 0C 0C 00 0C 0C 04 08 00 00 00 00 00 00 00
3C0 04 08 10 20 10 08 04 00 00 00 00 00 00 00 00 00
3D0 00 00 3E 00 3E 00 00 00 00 00 00 00 00 00 00 00
3E0 10 08 04 02 04 08 10 00 00 00 00 00 00 00 00 00
```

```
3F0 1C 22 02 04 08 08 00 08 00 00 00 00 00 00 00 00
400 3C 42 5A 56 5C 40 3E 00 00 00 00 00 00 00 00 00
410 18 24 42 42 7E 42 42 00 00 00 00 00 00 00 00 00
420 7C 22 22 3C 22 22 7C 00 00 00 00 00 00 00 00 00
430 3C 42 40 40 40 42 3C 00 00 00 00 00 00 00 00 00
440 7C 22 22 22 22 22 7C 00 00 00 00 00 00 00 00 00
450 7E 40 40 78 40 40 7E 00 00 00 00 00 00 00 00 00
460 7E 40 40 78 40 40 40 00 00 00 00 00 00 00 00 00
470 3C 40 40 4E 42 42 3C 00 00 00 00 00 00 00 00 00
480 42 42 42 7E 42 42 42 00 00 00 00 00 00 00 00 00
490 1C 08 08 08 08 08 1C 00 00 00 00 00 00 00 00 00
4A0 07 02 02 02 02 42 3C 00 00 00 00 00 00 00 00 00
4B0 42 44 48 70 48 44 42 00 00 00 00 00 00 00 00 00
4C0 40 40 40 40 40 42 7E 00 00 00 00 00 00 00 00 00
4D0 41 63 55 49 41 41 41 00 00 00 00 00 00 00 00 00
4E0 42 62 52 4A 46 42 42 00 00 00 00 00 00 00 00 00
4F0 3C 42 42 42 42 42 3C 00 00 00 00 00 00 00 00 00
500 7C 42 42 7C 40 40 40 00 00 00 00 00 00 00 00 00
510 3C 42 42 52 4A 44 3A 01 00 00 00 00 00 00 00 00
520 7C 42 42 7C 48 44 42 00 00 00 00 00 00 00 00 00
530 3E 40 40 3C 02 02 7C 00 00 00 00 00 00 00 00 00
540 7F 49 08 08 08 08 08 00 00 00 00 00 00 00 00 00
550 42 42 42 42 42 42 3C 00 00 00 00 00 00 00 00 00
560 41 41 22 22 14 14 08 00 00 00 00 00 00 00 00 00
570 41 41 41 49 55 63 41 00 00 00 00 00 00 00 00 00
580 41 22 14 08 14 22 41 00 00 00 00 00 00 00 00 00
590 41 22 14 08 08 08 08 00 00 00 00 00 00 00 00 00
5A0 7E 04 08 10 20 40 7E 00 00 00 00 00 00 00 00 00
5B0 1C 10 10 10 10 10 1C 00 00 00 00 00 00 00 00 00
5C0 40 20 10 08 04 02 01 00 00 00 00 00 00 00 00 00
5D0 1C 04 04 04 04 04 1C 00 00 00 00 00 00 00 00 00
5E0 08 1C 2A 08 08 08 08 00 00 00 00 00 00 00 00 00
5F0 00 00 00 00 00 00 7E 00 00 00 00 00 00 00 00 00
600 08 04 04 00 00 00 00 00 00 00 00 00 00 00 00 00
610 00 00 3C 02 3E 42 3E 00 00 00 00 00 00 00 00 00
620 40 40 5C 62 42 42 7C 00 00 00 00 00 00 00 00 00
630 00 00 3E 40 40 40 3E 00 00 00 00 00 00 00 00 00
640 02 02 3A 46 42 42 3E 00 00 00 00 00 00 00 00 00
650 00 00 3C 42 7E 40 3E 00 00 00 00 00 00 00 00 00
660 1C 22 20 70 20 20 20 00 00 00 00 00 00 00 00 00
670 00 00 3C 42 42 42 3E 02 3C 00 00 00 00 00 00 00
680 40 40 5C 62 42 42 42 00 00 00 00 00 00 00 00 00
690 08 00 08 08 08 08 08 00 00 00 00 00 00 00 00 00
6A0 02 00 02 02 02 02 42 24 18 00 00 00 00 00 00 00
6B0 40 40 46 58 60 58 46 00 00 00 00 00 00 00 00 00
6C0 08 08 08 08 08 08 08 00 00 00 00 00 00 00 00 00
6D0 00 00 56 69 49 49 49 00 00 00 00 00 00 00 00 00
6E0 00 00 5C 62 42 42 42 00 00 00 00 00 00 00 00 00
6F0 00 00 3C 42 42 42 3C 00 00 00 00 00 00 00 00 00
700 00 00 5C 62 42 62 5C 40 40 00 00 00 00 00 00 00
710 00 00 3E 42 42 42 3E 02 01 00 00 00 00 00 00 00
720 00 00 5C 62 40 40 40 00 00 00 00 00 00 00 00 00
730 00 00 3E 40 3C 02 7C 00 00 00 00 00 00 00 00 00
740 08 08 3E 08 08 08 08 00 00 00 00 00 00 00 00 00
750 00 00 42 42 42 46 3A 00 00 00 00 00 00 00 00 00
760 00 00 41 41 22 14 08 00 00 00 00 00 00 00 00 00
770 00 00 49 49 49 49 36 00 00 00 00 00 00 00 00 00
780 00 00 22 14 08 14 22 00 00 00 00 00 00 00 00 00
790 00 00 42 42 42 46 3A 02 3C 00 00 00 00 00 00 00
7A0 00 00 7E 04 18 20 7E 00 00 00 00 00 00 00 00 00
7B0 04 08 08 10 08 08 04 00 00 00 00 00 00 00 00 00
7C0 08 08 08 08 08 08 08 08 08 00 00 00 00 00 00 00
7D0 10 08 08 04 08 08 10 00 00 00 00 00 00 00 00 00
7E0 00 00 11 2A 44 00 00 00 00 00 00 00 00 00 00 00
7F0 55 2A 55 2A 55 0A 55 00 00 00 00 00 00 00 00 00
```

ROM CONTENTS



ASSEMBLY DRAWING

VB3

6x7 CHARACTER EPROM
RETROFIT
Part No. MN0027

VB3
VIDEO INTERFACE SOFTWARE
VERSION 2.0

INSTRUCTION MANUAL

AUGUST 1981

SSM Microcomputer Products, Inc.
2190 Paragon Drive
San Jose, California 95131

TELEPHONE: (408) 946-7400
TWX: 910-338-2077
TELEX: 171171
DDD: (408) 946-3644 (110 BAUD)

Part Number: MN0024

TABLE OF CONTENTS

Written By: Malcolm T. Wright & Dan Fischler

CP/M, MAC, and ASM are registered trademarks of Digital Research, P.O. Box 579, Pacific Grove, California 93950

8080 and 8085 are registered trademarks of Intel Corp., 3065 Bowers Avenue, Santa Clara, California 95051

Z80 is a registered trademark of Zilog Inc., 10340 Bubb Road, Cupertino, California 95014

Word Star is a trademark of Micro Pro International Corp., 1299 Fourth Street, San Rafael, California 94901

TeleVideo Systems, Inc., 1170 Morse Avenue, Sunnyvale, California 94086

VB3 is a trademark of SSM Microcomputer Products, Inc., 2190 Paragon Drive, San Jose, California 95131

1.0 VB3 Test & Initialization Routine

"VZ" is a special routine for initializing the VB3 to display different screen formats as defined by the user. VZ provides menu's to help the user specify the operating parameters for the VB3 including number of lines, top margin, horizontal position, etc. The user should read the VB3 manual for some basic understanding of the board's operation.

To execute the VZ routine, type "VZ" followed by a carriage return. After the CP/M prompt, for example, if the program was on drive A, the following sequence would execute VZ:

NOTE: <CR> = Carriage Return

A>VZ<CR>

VZ internally has the starting memory address of the VB3 character and attribute memory set for C000 (HEX), but this can be set by the user to any desired 8K boundary, when the user is asked at the beginning of the routine. The keyboard bank switch port is preset to E0 and E1 on start-up of VZ but also can be changed by the user at any time by selection of the proper menu number. This feature is useful if your system has multiple VB3's.

If item number 1 (USER DEFINED INITIALIZATION) in the main menu is continually picked and only one parameter is being changed each time, a carriage return can be used for the other parameters that will not be changed. In other words, if you don't want to change any particular parameter, just enter a carriage return.

Item numbers 2 (PRINT USER INITIALIZATION) and 5 (PRINT OUT VB3 VALUES) of the menu are valid only after the format information has been sent to the VB3.

The test pattern that is sent to the VB3 from VZ is comprised of a title and then multiple patterns of two lines of 128 characters. Each set of 256 characters will have one particular combination of attribute codes set in the attribute byte in the VB3 memory. The following combinations of attributes will be displayed:

Title
Alpha Characters
Inverted Characters
Flashing Characters
Inverted Gray Characters
Underlined Characters
Graphic Patterns
Inverted Flashing Characters
Thin-Line Graphics
Strike-Thru Characters
Blanked-Out Characters
Inverted Flashing Gray Characters
Alpha Characters
Gray Characters

Flashing Characters
and Spare Lines

2.0 Simple VB3 Driver for CP/M

"VBIOS" consists of four driver routines that can be used to interface the VB3 to CP/M (or any equivalent operating system). The entry point for each routine is a jump instruction in a "Jump Table" as follows:

```
JMP VBINIT
JMP VBIN
JMP VBOUT
JMP VBSTAT
```

VBINIT is the routine to initialize the VB3 and should be done only once during the "COLD BOOT" of CP/M into the computer. A "CALL" instruction to this label should be added to the cold boot routine.

VBIN is the routine to input data from the VB3's keyboard input. If your main console's keyboard is plugged into the VB3, then VBIN = Console Input for CP/M.

VBOUT is the routine to output data in Register C to the VB3's display. If your main console's display will be from the VB3, then VBOUT = Console Output in CP/M.

VBSTAT is the routine to test the data-available condition of the VB3's input. If you are using the VBIN in your BIOS, then VBSTAT = Console Input Status.

VBIOSM.MAC is written to assemble under "MAC" (the Digital Research Macro Assembler) and will require you to define a library of parameters to be used for VB3 initialization. This disk has some preset libraries called X80Xxx.LIB, where xx is the number of horizontal lines desired to appear on the screen. If MAC is not used, then the file VBIOS.ASM should be used on systems using either the 8080 or 8085. For systems using the Z80, use the file VBIOSZ.Z80.

The only commands that have been implemented with VBIOS are carriage return, line-feed, and backspace.

3.0 Library of Parameters

To define the operation of the VB3 board, example parameter libraries have been placed on this disk to set formats of 80 by 24, 80 by 33, etc. These libraries are used by MAC during the assembly of VBIOSM.MAC or ICRTM.MAC.

4.0 Intelligent Terminal Routine

The "ICRT" routine gives the VB3 the power of a good word processing terminal, with many special commands for screen editing. ICRT is about 1.5K bytes in size and should be located in memory above CP/M.

The file ICRTM.MAC is written for assembly with MAC. The file ICRTA.ASM is written for assembly using the standard CP/M assembler, ASM, on systems utilizing the 8080 or 8085. The file ICRTZ.Z80 should be used on systems operating with the Z80.

5.0 Televideo Simulator Routine

The "TELxx.ASM" routines provide the VB3 user with a driver whose control and escape sequences are compatible with the Televideo line of terminals. This is very useful when operating with software (i.e. Word Star) that is written to use these sequences. By using the Televideo-like control sequences, interfaceing the VB3 to an applications program such as Word Star is greatly simplified with very little or no "patching" needed.

6.0 User Supplied Routines

VB3GRAPH is a collection of three subroutines that can be used in a user-defined program to read, write, and verify the coarse graphic squares generated by the VB3. The entire screen is treated as an XY matrix with the lower left-hand corner equal to coordinates 0,0. The horizontal resolution is 160 squares (80 x 2). The vertical resolution is defined as NROW x 4 squares. NROW must be equated to the number of lines set by the user during VB3 initialization.

The VB3 must have all of the attribute bytes in the VB3 memory set to graphics. Also, the bottom line on the screen must be set to NROW-1 for correct operation. The entry points for the CALLed subroutines are designated CHECK, WHITE, and BLACK. The XY coordinates are passed to the subroutines as contents of Registers H and L (X coordinate:L=0 to 159; Y coordinate:H=0 to (NROW-1)x4). No hardware scrolling commands should be sent to the VB3 during graphics or the matrix will become discontinuous and not display correctly.

WINDOWS is a routine which treats the VB3 (actually works with virtually any memory mapped video board) as two or more smaller video screens. Each smaller screen will accept commands like carriage return, line-feed, backspace, home, up, down, right, left, and clear screen. One main entry point labeled ENTER allows access to any of the smaller screens by passing the screen number as the contents of register A. (Additional information is given in the source listing of WINDOWS.ASM).

7.0 Solicitation of Additional User Supplied Routines

If you would like any of your VB3 programs to be included on this disk in the future, please submit them to the Customer Service Department at SSM.

STANDARD MICROSYSTEMS CORPORATION

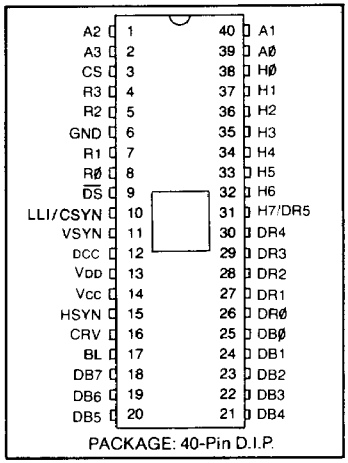
CRT 5027
CRT 5037
CRT 5057
μPC FAMILY

CRT Video Timer and Controller
VTAC®

FEATURES

- Fully Programmable Display Format
  - Characters per data row (1-200)
  - Data rows per frame (1-64)
  - Raster scans per data row (1-16)
- Programmable Monitor Sync Format
  - Raster Scans/Frame (256-1023)
  - "Front Porch"
  - Sync Width
  - "Back Porch"
  - Interlace/Non-Interlace
  - Vertical Blanking
- Lock Line Input (CRT 5057)
- Direct Outputs to CRT Monitor
  - Horizontal Sync
  - Vertical Sync
  - Composite Sync (CRT 5027, CRT 5037)
  - Blanking
  - Cursor coincidence
- Programmed via:
  - Processor data bus
  - External PROM
  - Mask Option ROM
- Standard or Non-Standard CRT Monitor Compatible
- Refresh Rate: 60Hz, 50Hz,...
- Scrolling
  - Single Line
  - Multi-Line
- Cursor Position Registers
- Character Format: 5x7, 7x9,...
- Programmable Vertical Data Positioning
- Balanced Beam Current Interlace (CRT 5037)
- Graphics Compatible

PIN CONFIGURATION



PACKAGE: 40-Pin D.I.P.

- Split-Screen Applications
  - Horizontal
  - Vertical
- Interlace or Non-Interlace operation
- TTL Compatibility
- BUS Oriented
- High Speed Operation
- COPLAMOS® N-Channel Silicon Gate Technology
- Compatible with CRT 8002 VDAC™
- Compatible with CRT 7004

GENERAL DESCRIPTION

The CRT Video Timer and Controller Chip (VTAC)* is a user programmable 40-pin COPLAMOS® n channel MOS/LSI device containing the logic functions required to generate all the timing signals for the presentation and formatting of interlaced and non-interlaced video data on a standard or non-standard CRT monitor.

With the exception of the dot counter, which may be clocked at a video frequency above 25 MHz and therefore not recommended for MOS implementation, all frame formatting, such as horizontal, vertical, and composite sync, characters per data row, data rows per frame, and raster scans per data row and per frame are totally user programmable. The data row counter has been designed to facilitate scrolling.

Programming is effected by loading seven 8 bit control registers directly off an 8 bit bidirectional data bus. Four register address lines and a chip select line provide complete microprocessor compatibility for program controlled set up. The device can be "self loaded" via an external PROM tied on the data bus as described in the OPERATION section. Formatting can also be programmed by a single mask option.

In addition to the seven control registers two additional registers are provided to store the cursor character and data row addresses for generation of the cursor video signal. The contents of these two registers can also be read out onto the bus for update by the program.

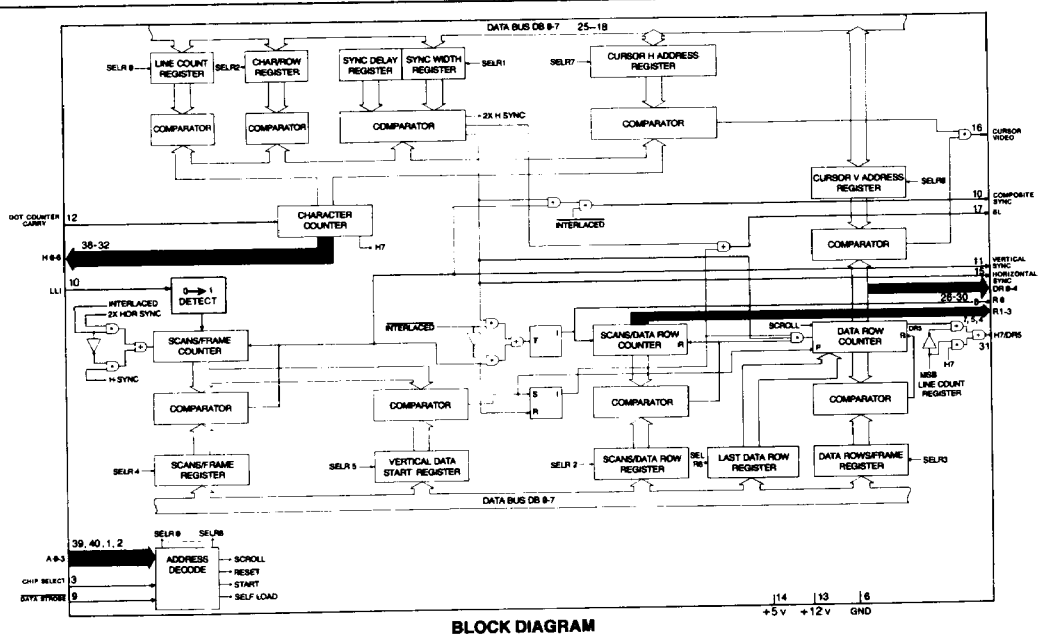
Three versions of the VTAC® are available. The CRT 5027 provides non-interlaced operation with an even or odd number of scan lines per data row, or interlaced operation with an even number of scan lines per data row. The CRT 5037 may be programmed for an odd or even number of scan lines per data row in both interlaced and non-interlaced modes. Programming the CRT 5037 for an odd number of scan lines per data row eliminates character distortion caused by the uneven beam current normally associated with odd field/even field interlacing of alphanumeric displays.

The CRT 5057 provides the ability to lock a CRT's vertical refresh rate, as controlled by the VTAC's® vertical sync pulse, to the 50 Hz or 60 Hz line frequency thereby eliminating the so called "swim" phenomenon. This is particularly well suited for European system requirements. The line frequency waveform, processed to conform to the VTAC's® specified logic levels, is applied to the line lock input. The VTAC® will inhibit generation of vertical sync until a zero to one transition on this input is detected. The vertical sync pulse is then initiated within one scan line after this transition rises above the logic threshold of the VTAC.®

To provide the pin required for the line lock input, the composite sync output is not provided in the CRT 5057.

Description of Pin Functions

Pin No.	Symbol	Name	Input/ Output	Function
25-18	DBØ-7	Data Bus	I/O	Data bus. Input bus for control words from microprocessor or PROM. Bidirectional bus for cursor address.
3	CS	Chip Select	I	Signals chip that it is being addressed
39,40,1,2	AØ-3	Register Address	I	Register address bits for selecting one of seven control registers or either of the cursor address registers
9	$\overline{DS}$	Data Strobe	I	Strobes DBØ-7 into the appropriate register or outputs the cursor character address or cursor line address onto the data bus
12	DCC	DOT Counter Carry	I	Carry from off chip dot counter establishing basic character clock rate. Character clock.
38-32	HØ-6	Character Counter Outputs	O	Character counter outputs.
7, 5, 4	R1-3	Scan Counter Outputs	O	Three most significant bits of the Scan Counter; row select inputs to character generator.
31	H7/DR5	H7/DR5	O	Pin definition is user programmable. Output is MSB of Character Counter if horizontal line count (REG.Ø) is ≥128; otherwise output is MSB of Data Row Counter.
8	RØ	Scan Counter LSB	O	Least significant bit of the scan counter. In the interlaced mode with an even number of scans per data row, RØ will toggle at the field rate; for an odd number of scans per data row in the interlaced mode, RØ will toggle at the data row rate.
26-30	DRØ-4	Data Row Counter Outputs	O	Data Row counter outputs.
17	BL	Blank	O	Defines non active portion of horizontal and vertical scans.
15	HSYN	Horizontal Sync	O	Initiates horizontal retrace.
11	VSYN	Vertical Sync	O	Initiates vertical retrace.
10	CSYN/ LLI	Composite Sync Output/ Line Lock Input	O/I	Composite sync is provided on the CRT 5027 and CRT 5037. This output is active in non-interlaced mode only. Provides a true RS-170 composite sync wave form. For the CRT 5057, this pin is the Line Lock Input. The line frequency waveform, processed to conform to the VTAC's® specified logic levels, is applied to this pin.
16	CRV	Cursor Video	O	Defines cursor location in data field.
14	V_{CC}	Power Supply	PS	+5 volt Power Supply
13	V_{DD}	Power Supply	PS	+12 volt Power Supply



BLOCK DIAGRAM

Operation

The design philosophy employed was to allow the device to interface effectively with either a microprocessor based or hardwire logic system. The device is programmed by the user in one of two ways; via the processor data bus as part of the system initialization routine, or during power up via a PROM tied on the data bus and addressed directly by the Row Select outputs of the chip. (See figure 4). Seven 8 bit words are required to fully program the chip. Bit assignments for these words are shown in Table 1. The information contained in these seven words consists of the following:

Horizontal Formatting:

Characters/Data Row	A 3 bit code providing 8 mask programmable character lengths from 20 to 132. The standard device will be masked for the following character lengths; 20, 32, 40, 64, 72, 80, 96, and 132.
Horizontal Sync Delay	3 bits assigned providing up to 8 character times for generation of "front porch".
Horizontal Sync Width	4 bits assigned providing up to 15 character times for generation of horizontal sync width.
Horizontal Line Count	8 bits assigned providing up to 256 character times for total horizontal formatting.
Skew Bits	A 2 bit code providing from a 0 to 2 character skew (delay) between the horizontal address counter and the blank and sync (horizontal, vertical, composite) signals to allow for retiming of video data prior to generation of composite video signal. The Cursor Video signal is also skewed as a function of this code.

Vertical Formatting:

Interlaced/Non-interlaced	This bit provides for data presentation with odd/even field formatting for interlaced systems. It modifies the vertical timing counters as described below. A logic 1 establishes the interlace mode.
Scans/Frame	8 bits assigned, defined according to the following equations: Let X = value of 8 assigned bits. 1) in interlaced mode—scans/frame = 2X + 513. Therefore for 525 scans, program X = 6 (00000110). Vertical sync will occur precisely every 262.5 scans, thereby producing two interlaced fields. Range = 513 to 1023 scans/frame, odd counts only. 2) in non-interlaced mode—scans/frame = 2X + 256. Therefore for 262 scans, program X = 3 (00000011). Range = 256 to 766 scans/frame, even counts only. In either mode, vertical sync width is fixed at three horizontal scans (= 3H).
Vertical Data Start	8 bits defining the number of raster scans from the leading edge of vertical sync until the start of display data. At this raster scan the data row counter is set to the data row address at the top of the page.
Data Rows/Frame	6 bits assigned providing up to 64 data rows per frame.
Last Data Row	6 bits to allow up or down scrolling via a preload defining the count of the last displayed data row.
Scans/Data Row	4 bits assigned providing up to 16 scan lines per data row.

Additional Features

Device Initialization:

Under microprocessor control—The device can be reset under system or program control by presenting a 1Ø1Ø address on A3-Ø. The device will remain reset at the top of the even field page until a start command is executed by presenting a 111Ø address on A3-Ø.

Via "Self Loading"—In a non-processor environment, the self loading sequence is effected by presenting and holding the 1111 address on A3-Ø, and is initiated by the receipt of the strobe pulse ($\overline{DS}$). The 1111 address should be maintained long enough to insure that all seven registers have been loaded (in most applications under one millisecond). The timing sequence will begin one line scan after the 1111 address is removed. In processor based systems, self loading is initiated by presenting the Ø111 address to the device. Self loading is terminated by presenting the start command to the device which also initiates the timing chain.

Scrolling—In addition to the Register 6 storage of the last displayed data row a "scroll" command (address 1Ø11) presented to the device will increment the first displayed data row count to facilitate up scrolling in certain applications.

Control Registers Programming Chart

Horizontal Line Count: Total Characters/Line = N + 1, N = 0 to 255 (DB0 = LSB)

Characters/Data Row:

DB2	DB1	DB0		
0	0	0	= 20	Active Characters/Data Row
0	0	1	= 32	
0	1	0	= 40	
0	1	1	= 64	
1	0	0	= 72	
1	0	1	= 80	
1	1	0	= 96	
1	1	1	= 132	

Horizontal Sync Delay: = N, from 1 to 7 character times (DB0 = LSB) (N = 0 Disallowed)

Horizontal Sync Width: = N, from 1 to 15 character times (DB3 = LSB) (N = 0 Disallowed)

Skew Bits

DB7	DB6	Sync/Blank Delay (Character Times)	Cursor Delay (Character Times)
0	0	0	0
1	0	1	0
0	1	2	1
1	1	2	2

Scans/Frame

8 bits assigned, defined according to the following equations:
Let X = value of 8 assigned bits. (DB0 = LSB)

1) in interlaced mode—scans/frame = 2X + 513. Therefore for 525 scans, program X = 6 (00000110). Vertical sync will occur precisely every 262.5 scans, thereby producing two interlaced fields.

Range = 513 to 1023 scans/frame, odd counts only.

2) in non-interlaced mode—scans/frame = 2X + 256. Therefore for 262 scans, program X = 3 (00000011).

Range = 256 to 766 scans/frame, even counts only.

In either mode, vertical sync width is fixed at three horizontal scans (= 3H).

Vertical Data Start: N = number of raster lines delay after leading edge of vertical sync of vertical start position. (DB0 = LSB)

Data Rows/Frame: Number of data rows = N + 1, N = 0 to 63 (DB0 = LSB)

Last Data Row: N = Address of last dsplayed data row, N = 0 to 63, ie; for 24 data rows, program N = 23. (DB0 = LSB)

Mode: Register, 1, DB7 = 1 establishes Interlace.

Scans/Data Row:

Interlace Mode

CRT 5027: Scans per Data Row = N + 1 where N = programmed number of scans/data rows. N = 0 to 15. Scans per data row must be even counts only.

CRT 5037, CRT 5057: Scans per data Row = N + 2. N = 0 to 14, odd or even counts.

Non-Interlace Mode

CRT 5027, CRT 5037, CRT 5057: Scans per Data Row = N + 1, odd or even count. N = 0 to 15.

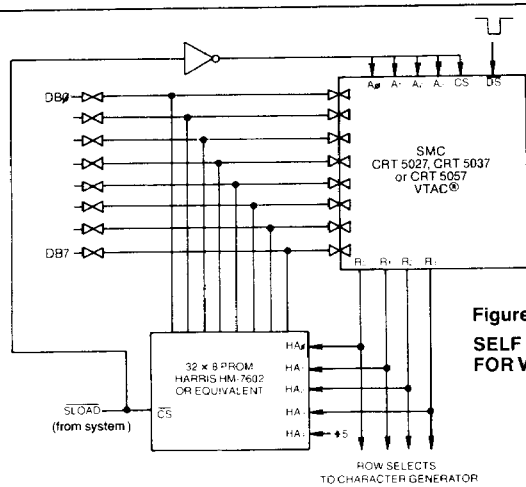


Figure 4.

SELF LOADING SCHEME FOR VTAC® SET-UP

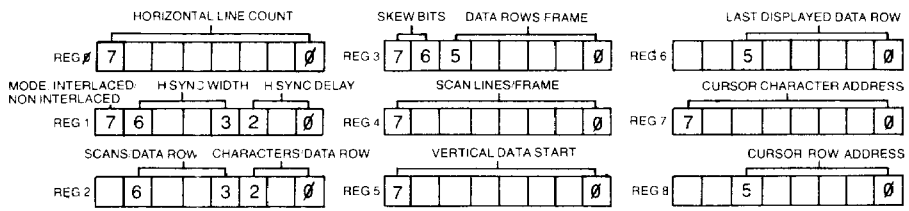
Register Selects/Command Codes

A3	A2	A1	AØ	Select/Command	Description
0	0	0	0	Load Control Register Ø	See Table 1
0	0	0	1	Load Control Register 1	
0	0	1	0	Load Control Register 2	
0	0	1	1	Load Control Register 3	
0	1	0	0	Load Control Register 4	
0	1	0	1	Load Control Register 5	
0	1	1	0	Load Control Register 6	
0	1	1	1	Processor Initiated Self Load	Command from processor instructing VTAC® to enter Self Load Mode (via external PROM)
1	0	0	0	Read Cursor Line Address	
1	0	0	1	Read Cursor Character Address	
1	0	1	0	Reset	Resets timing chain to top left of page. Reset is latched on chip by $\overline{DS}$ and counters are held until released by start command.
1	0	1	1	Up Scroll	Increments address of first displayed data row on page. ie; prior to receipt of scroll command—top line = 0, bottom line = 23. After receipt of Scroll Command—top line = 1, bottom line = 0.
1	1	0	0	Load Cursor Character Address*	
1	1	0	1	Load Cursor Line Address*	
1	1	1	0	Start Timing Chain	Receipt of this command after a Reset or Processor Self Load command will release the timing chain approximately one scan line later. In applications requiring synchronous operation of more than one CRT 5027 the dot counter carry should be held low during the $\overline{DS}$ for this command.
1	1	1	1	Non-Processor Self Load	Device will begin self load via PROM when $\overline{DS}$ goes low. The 1111 command should be maintained on A3-Ø long enough to guarantee self load. (Scan counter should cycle through at least once). Self load is automatically terminated and timing chain initiated when the all "1's" condition is removed, independent of $\overline{DS}$. For synchronous operation of more than one VTAC®, the Dot Counter Carry should be held low when the command is removed.

*NOTE: During Self-Load, the Cursor Character Address Register (REG 7) and the Cursor Row Address Register (REG 8) are enabled during states Ø111 and 1ØØØ of the R3-RØ Scan Counter outputs respectively. Therefore, Cursor data in the PROM should be stored at these addresses.

TABLE 1

BIT ASSIGNMENT CHART



SECTION V

AC TIMING DIAGRAMS

FIGURE 1 VIDEO TIMING

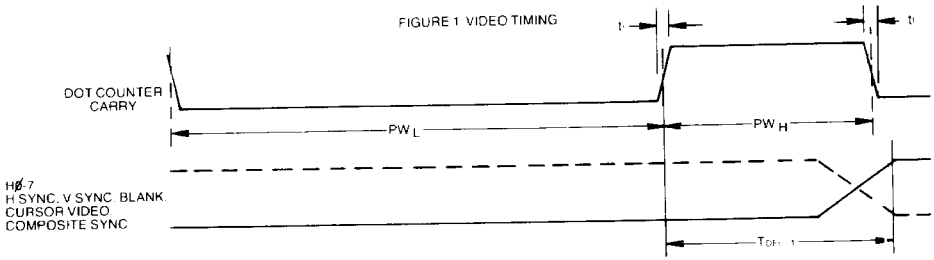


FIGURE 2 LOAD READ TIMING

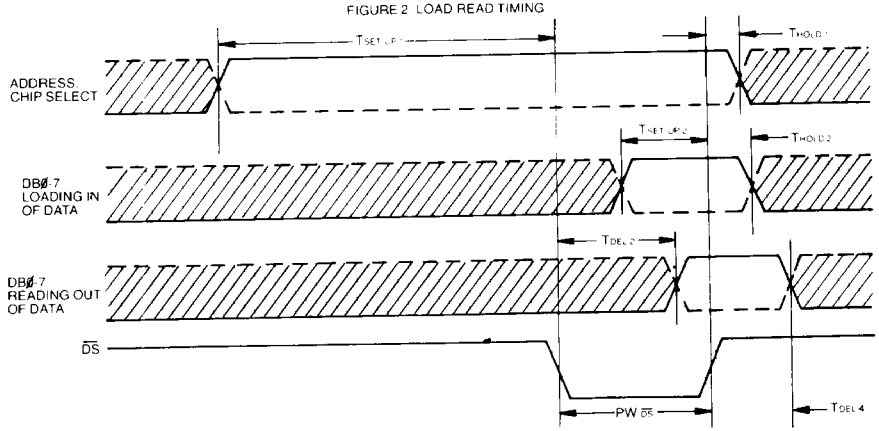
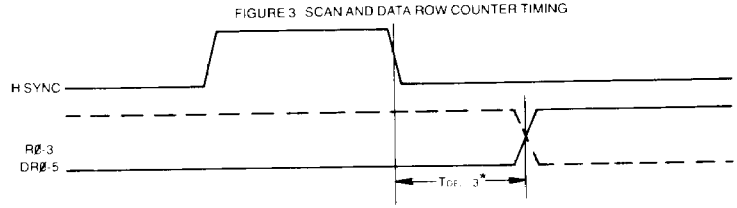
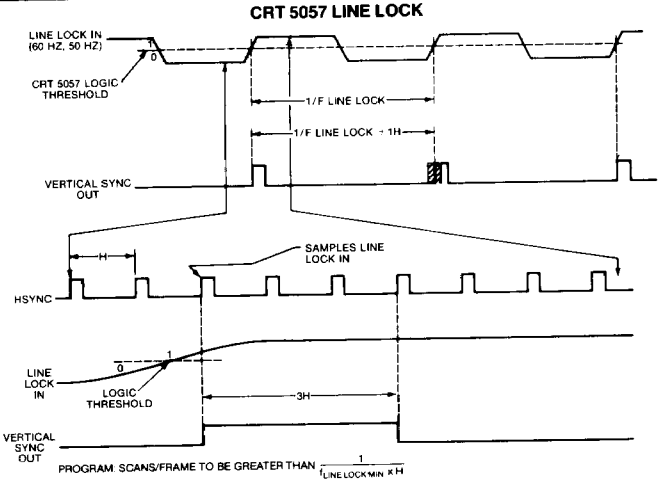


FIGURE 3 SCAN AND DATA ROW COUNTER TIMING



*RØ-3 and DRØ-5 may change prior to the falling edge of H sync

CRT 5057 LINE LOCK



Note: To ensure a stable display when using the line lock mode, the CRT 5057 should be used with DC coupled monitors only.

MAXIMUM GUARANTEED RATINGS*

Operating Temperature Range	0°C to + 70°C
Storage Temperature Range	−55°C to +150°C
Lead Temperature (soldering, 10 sec.)	+325°C
Positive Voltage on any Pin, with respect to ground	+18.0V
Negative Voltage on any Pin, with respect to ground	−0.3V

*Stresses above those listed may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operational sections of this specification is not implied.

NOTE: When powering this device from laboratory or system power supplies, it is important that the Absolute Maximum Ratings not be exceeded or device failure can result. Some power supplies exhibit voltage spikes or "glitches" on their outputs when the AC power is switched on and off. In addition, voltage transients on the AC power line may appear on the DC output. For example, the bench power supply programmed to deliver +12 volts may have large voltage transients when the AC power is switched on and off. If this possibility exists it is suggested that a clamp circuit be used.

ELECTRICAL CHARACTERISTICS (T_A=0°C to 70°C, V_{CC} = +5V±5%, V_{DD} = −12V±5%, unless otherwise noted)

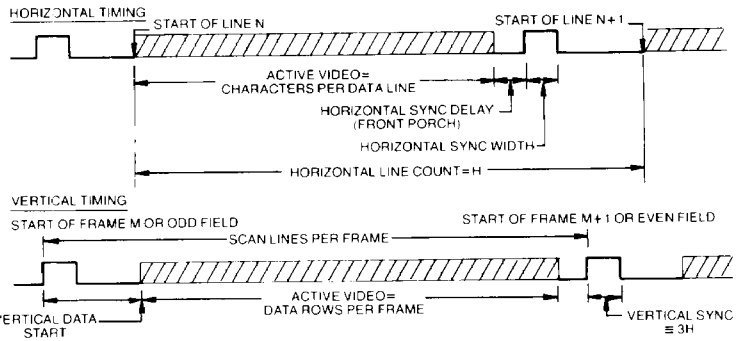
Parameter	Min.	Typ.	Max.	Unit	Comments
D.C. CHARACTERISTICS					
INPUT VOLTAGE LEVELS					
Low Level, V_{IL}			0.8	V	
High Level, V_{IH}	V_{CC}−1.5		V_{CC}	V	
OUTPUT VOLTAGE LEVELS					
Low Level—V_{OL} for RØ-3			0.4	V	I_{OL} = 3.2ma
Low Level—V_{OL} all others			0.4	V	I_{OL} = 1.6ma
High Level—V_{OH} for RØ-3, DBØ-7	2.4				I_{OH} = 80μa
High Level—V_{OH} all others	2.4				I_{OH} = 40μa
INPUT CURRENT					
Low Level, I_{IL} (Address, CS only)			250	μA	V_{IN} = 0.4V
Leakage, I_{IL} (All Inputs except Address, CS)			10	μA	$O \leq V_{IN} \leq V_{CC}$
INPUT CAPACITANCE					
Data Bus, C_{IN}		10	15	pF	
$\overline{DS}$, Clock, C_{IN}		25	40	pF	
All other, C_{IN}		10	15	pF	
DATA BUS LEAKAGE in INPUT MODE					
I_{DB}			10	μA	$0.4V \leq V_{IN} \leq 5.25V$
POWER SUPPLY CURRENT					
I_{CC}		80	100	mA	
I_{DD}		40	70	mA	
A.C. CHARACTERISTICS					T_A = 25°C
DOT COUNTER CARRY					
frequency	0.5		4.0	MHz	Figure 1
PW_H	35			ns	Figure 1
PW_L	215			ns	Figure 1
t_r, t_f		10	50	ns	Figure 1
DATA STROBE					
$PW_{\overline{DS}}$	150ns		10μs		Figure 2
ADDRESS, CHIP SELECT					
Set-up time	125			ns	Figure 2
Hold time	50			ns	Figure 2
DATA BUS—LOADING					
Set-up time	125			ns	Figure 2
Hold time	75			ns	Figure 2
DATA BUS—READING					
T_{DEL2}			125	ns	Figure 2, CL=50pF
T_{DEL4}	5		60	ns	Figure 2, CL=50pF
OUTPUTS: HØ-7, HS, VS, BL, CRV, CS-T_{DEL1}			125	ns	Figure 1, CL=20pF
OUTPUTS: RØ-3, DRØ-5					
T_{DEL3}	*		750	ns	Figure 3, CL=20pF

*RØ-3 and DRØ-5 may change prior to the falling edge of H sync

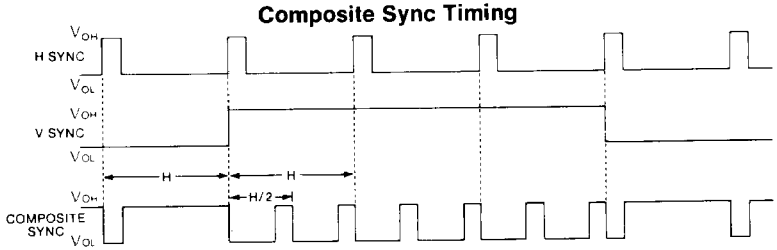
Restrictions

1. Only one pin is available for strobing data into the device via the data bus. The cursor X and Y coordinates are therefore loaded into the chip by presenting one set of addresses and outputed by presenting a different set of addresses. Therefore the standard WRITE and READ control signals from most microprocessors must be "NORed" externally to present a single strobe ($\overline{DS}$) signal to the device.
2. In interlaced mode the total number of character slots assigned to the horizontal scan must be even to insure that vertical sync occurs precisely between horizontal sync pulses.

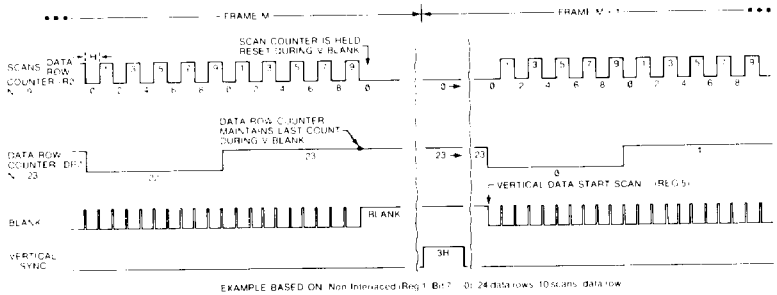
General Timing



Composite Sync Timing



Vertical Sync Timing



Start-up, CRT 5027

When employing microprocessor controlled loading of the CRT 5027's registers, the following sequence of instructions is necessary:

ADDRESS				COMMAND
1	1	1	0	Start Timing Chain
1	0	1	0	Reset
0	0	0	0	Load Register 0
		•		•
		•		•
		•		•
0	1	1	0	Load Register 6
1	1	1	0	Start Timing Chain

The sequence of START RESET LOAD START is necessary to insure proper initialization of the registers.

This sequence is not required if register loading is via either of the Self Load modes. This sequence is optional with the CRT 5037 or CRT 5057.

STANDARD MICROSYSTEMS CORPORATION

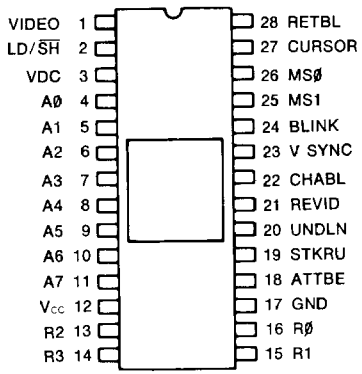
CRT 8002
μPC FAMILY

CRT Video Display Attributes Controller Video Generator VDAC™

FEATURES

- ☐ On chip character generator (mask programmable)
  - 128 Characters (alphanumeric and graphic)
  - 7 x 11 Dot matrix block
- ☐ On chip video shift register
  - Maximum shift register frequency
    - CRT 8002A 20MHz
    - CRT 8002B 15MHz
    - CRT 8002C 10MHz
    - Access time 400ns
- ☐ On chip horizontal and vertical retrace video blanking
- ☐ No descender circuitry required
- ☐ Four modes of operation (intermixable)
  - Internal character generator (ROM)
  - Wide graphics
  - Thin graphics
  - External inputs (fonts/dot graphics)
- ☐ On chip attribute logic—character, field
  - Reverse video
  - Character blank
  - Character blink
  - Underline
  - Strike-thru
- ☐ Four on chip cursor modes
  - Underline
  - Blinking underline
  - Reverse video
  - Blinking reverse video
- ☐ Programmable character blink rate
- ☐ Programmable cursor blink rate
- ☐ Subscriptable
- ☐ Expandable character set
  - External fonts
  - Alphanumeric and graphic
  - RAM, ROM, and PROM
- ☐ On chip address buffer
- ☐ On chip attribute buffer
- ☐ +5 volt operation
- ☐ TTL compatible
- ☐ MOS N-channel silicon-gate COPLAMOS® process
- ☐ CLASP® technology—ROM and options
- ☐ Compatible with CRT 5027 VTAC®

PIN CONFIGURATION



SECTION V

General Description

The SMC CRT 8002 Video Display Attributes Controller (VDAC) is an N-channel COPLAMOS® MOS/LSI device which utilizes CLASP® technology. It contains a 7X11X128 character generator ROM, a wide graphics mode, a thin graphics mode, an external input mode, character address/data latch, field and/or character attribute logic, attribute latch, four cursor modes, two programmable blink rates, and a high speed video shift register. The CRT 8002 VDAC™ is a companion chip to SMC's CRT 5027 VTAC. Together these two chips comprise the circuitry required for the display portion of a CRT video terminal.

The CRT 8002 video output may be connected directly to a CRT monitor video input. The CRT 5027 blanking output can be connected directly to the CRT 8002 retrace blank input to provide both horizontal and vertical retrace blanking of the video output.

Four cursor modes are available on the CRT 8002. They are: underline, blinking underline, reverse video block, and blinking reverse video block. Any one of these can be mask programmed as the cursor function. There is a separate cursor blink rate which can be mask programmed to provide a 15Hz to 1Hz blink rate.

The CRT 8002 attributes include: reverse video, character blank, blink, underline, and strike-thru. The character blink rate is mask programmable from 7.5Hz to 0.5Hz and has a duty cycle of 75/25. The underline and strike-thru are similar but independently controlled functions and can be mask programmed to any number of raster lines at any position in the character block. These attributes are available in all modes.

In the wide graphic mode the CRT 8002 produces a graphic entity the size of the character block. The graphic entity contains 8 parts, each of which is associated with one bit of a graphic byte, thereby providing for 256 unique graphic symbols. Thus, the CRT 8002 can produce either an alphanumeric symbol or a graphic entity depending on the mode selected. The mode can be changed on a per character basis.

The thin graphic mode enables the user to create single line drawings and forms.

The external mode enables the user to extend the on-chip ROM character set and/or the on-chip graphics capabilities by inserting external symbols. These external symbols can come from either RAM, ROM or PROM.

MAXIMUM GUARANTEED RATINGS*

Operating Temperature Range	0°C to + 70°C
Storage Temperature Range	−55°C to +150°C
Lead Temperature (soldering, 10 sec.)	+325°C
Positive Voltage on any Pin, with respect to ground	+8.0V
Negative Voltage on any Pin, with respect to ground	−0.3V

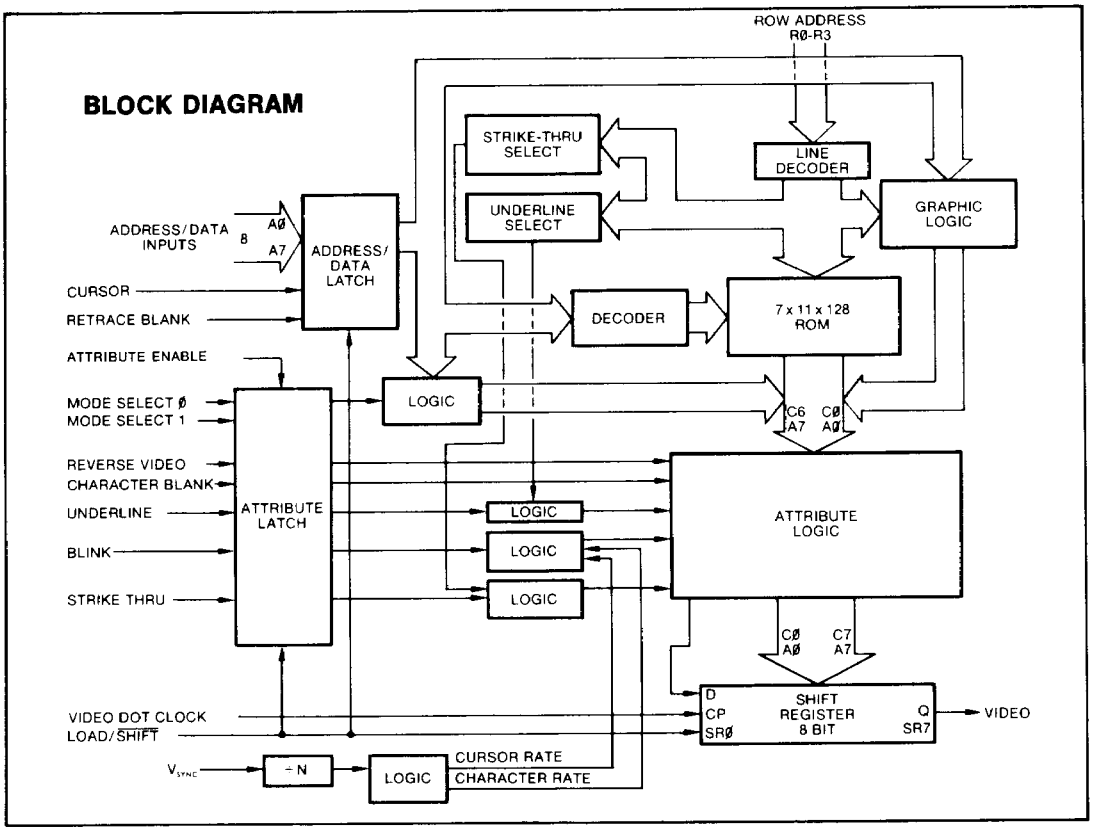
*Stresses above those listed may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operational sections of this specification is not implied.

NOTE: When powering this device from laboratory or system power supplies, it is important that the Absolute Maximum Ratings not be exceeded or device failure can result. Some power supplies exhibit voltage spikes or "glitches" on their outputs when the AC power is switched on and off. In addition, voltage transients on the AC power line may appear on the DC output. If this possibility exists it is suggested that a clamp circuit be used.

ELECTRICAL CHARACTERISTICS ($T_A = 0°C$ to $70°C$, $V_{CC} = +5V \pm 5\%$, unless otherwise noted)

Parameter	Min.	Typ.	Max.	Unit	Comments
D.C. CHARACTERISTICS					
INPUT VOLTAGE LEVELS					
Low-level, V_{IL}			0.8	V	excluding VDC
High-level, V_{IH}	2.0			V	excluding VDC
INPUT VOLTAGE LEVELS-CLOCK					
Low-level, V_{IL}			0.8	V	
High-level, V_{IH}	4.3			V	See Figure 6
OUTPUT VOLTAGE LEVELS					
Low-level, V_{OL}			0.4	V	$I_{OL} = 0.4$ mA, 74LSXX load
High-level, V_{OH}	2.4			V	$I_{OH} = -20\mu A$
INPUT CURRENT					
Leakage, I_L (Except CLOCK)			10	μA	$0 \leq V_{IN} \leq V_{CC}$
Leakage, I_L (CLOCK Only)			50	μA	$0 \leq V_{IN} \leq V_{CC}$
INPUT CAPACITANCE					
Data		10		pF	@ 1 MHz
LD/$\overline{SH}$		20		pF	@ 1 MHz
CLOCK		25		pF	@ 1 MHz
POWER SUPPLY CURRENT					
I_{CC}		100		mA	
A.C. CHARACTERISTICS					
See Figure 6, 7					

SYMBOL	PARAMETER	CRT 8002A		CRT 8002B		CRT 8002C		UNITS
		MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
VDC	Video Dot Clock Frequency	1.0	20	1.0	15	1.0	10	MHz
PW_H	VDC—High Time	15.0		23		40		ns
PW_L	VDC—Low Time	15.0		23		40		ns
t_{CY}	LD/$\overline{SH}$ cycle time	400		533		800		ns
t_r, t_f	Rise, fall time		10		10		10	ns
$t_{SET\text{-}UP}$	Input set-up time	≥0		≥0		≥0		ns
t_{HOLD}	Input hold time	15		15		15		ns
t_{PDI}, t_{PDO}	Output propagation delay	15	50	15	65	15	100	ns
t_1	LD/$\overline{SH}$ set-up time	10		15		20		ns
t_2	LD/$\overline{SH}$ hold time	15		15		15		ns



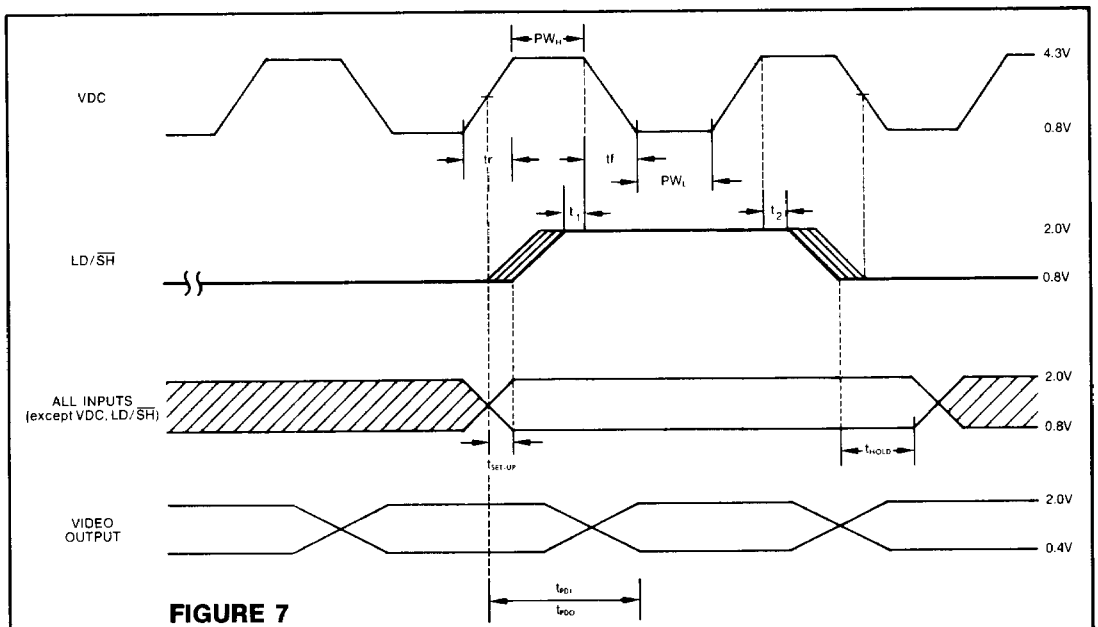


FIGURE 7
AC TIMING DIAGRAM

SECTION V

DESCRIPTION OF PIN FUNCTIONS

PIN NO.	SYMBOL	NAME	INPUT/ OUTPUT	FUNCTION
1	VIDEO	Video Output	O	The video output contains the dot stream for the selected row of the alpha-numeric, wide graphic, thin graphic, or external character after processing by the attribute logic, and the retrace blank and cursor inputs. In the alphanumeric mode, the characters are ROM programmed into the 77 dots, (7X11) allocated for each of the 128 characters. See figure 5. The top row (RØ) and rows R12 to R15 are normally all zeros as is column C7. Thus, the character is defined in the box bounded by R1 to R11 and CØ to C6. When a row of the ROM, via the attribute logic, is parallel loaded into the 8-bit shift-register, the first bit serially shifted out is C7 (A zero; or a one in REVID). It is followed by C6, C5, through CØ. The timing of the Load/$\overline{\text{Shift}}$ pulse will determine the number of additional (— —, zero to N) backfill zeros (or ones if in REVID) shifted out. See figure 4. When the next Load/$\overline{\text{Shift}}$ pulse appears the next character's row of the ROM, via the attribute logic, is parallel loaded into the shift register and the cycle repeats.
2	LD/$\overline{\text{SH}}$	Load/$\overline{\text{Shift}}$	I	The 8 bit shift-register parallel-in load or serial-out shift modes are established by the Load/$\overline{\text{Shift}}$ input. When low, this input enables the shift register for serial shifting with each Video Dot Clock pulse. When high, the shift register parallel (broadside) data inputs are enabled and synchronous loading occurs on the next Video Dot Clock pulse. During parallel loading, serial data flow is inhibited. The Address/Data inputs (AØ-A7) are latched on the negative transition of the Load/$\overline{\text{Shift}}$ input. See timing diagram, figure 7.
3	VDC	Video Dot Clock	I	Frequency at which video is shifted.
4-11	AØ-A7	Address/Data	I	In the Alphanumeric Mode the 7 bits on inputs (AØ-A6) are internally decoded to address one of the 128 available characters (A7 = X). In the External Mode, AØ-A7 is used to insert an 8 bit word from a user defined external ROM, PROM or RAM into the on-chip Attribute logic. In the wide Graphic Modes AØ-A7 is used to define one of 256 graphic entities. In the thin Graphic Mode AØ-A2 is used to define the 3 line segments.
12	V_{CC}	Power Supply	PS	+5 volt power supply
13,14,15,16	R2,R3,R1,RØ	Row Address	I	These 4 binary inputs define the row address in the current character block.
17	GND	Ground	GND	Ground
18	ATTBE	Attribute Enable	I	A positive level on this input enables data from the Reverse Video, Character Blank, Underline, Strike-Thru, Blink, Mode Select Ø, and Mode Select 1 inputs to be strobed into the on-chip attribute latch at the negative transition of the Load/$\overline{\text{Shift}}$ pulse. The latch loading is disabled when this input is low. The latched attributes will remain fixed until this input becomes high again. To facilitate attribute latching on a character by character basis, tie ATTBE high. See timing diagram, figure 7.
19	STKRU	Strike-Thru	I	When this input is high and RETBL = 0, the parallel inputs to the shift register are forced high (SRØ-SR7), providing a solid line segment throughout the character block. The operation of strike-thru is modified by Reverse Video (see table 1). In addition, an on-chip ROM programmable decoder is available to decode the line count on which strike-thru is to be placed as well as to program the strike-thru to be 1 to N raster lines high. Actually, the strike-thru decoder (mask programmable) logic allows the strike-thru to be any number or arrangement of horizontal lines in the character block. The standard strike-thru will be a double line on rows R5 and R6.
20	UNDLN	Underline	I	When this input is high and RETBL = 0, the parallel inputs to the shift register are forced high (SRØ-SR7), providing a solid line segment throughout the character block. The operation of underline is modified by Reverse Video (see table 1). In addition, an on-chip ROM programmable decoder is available to decode the line count on which underline is to be placed as well as to program the underline to be 1 to N raster lines high. Actually, the underline decoder (mask programmable) logic allows the underline to be any number or arrangement of horizontal lines in the character block. The standard underline will be a single line on R11.
21	REVID	Reverse Video	I	When this input is low and RETBL = 0, data into the Attribute Logic is presented directly to the shift register parallel inputs. When reverse video is high data into the Attribute Logic is inverted and then presented to the shift register parallel inputs. This operation reverses the data and field video. See table 1.
22	CHABL	Character Blank	I	When this input is high, the parallel inputs to the shift register are all set low, providing a blank character line segment. Character blank will override blink. The operation of Character Blank is modified by the Reverse Video input. See table 1.
23	V SYNC	V SYNC	I	This input is used as the clock input for the two on-chip mask programmable blink rate dividers. The cursor blink rate (50/50 duty cycle) will be twice the character blink rate (75/25 duty cycle). The divisors can be programmed from ÷ 4 to ÷ 30 for the cursor (÷ 8 to ÷ 60 for the character).
24	BLINK	Blink	I	When this input is high and RETBL = 0 and CHABL = 0, the character will blink at the programmed character blink rate. Blinking is accomplished by blanking the character block with the internal Character Blink clock. The standard character blink rate is 1.875 Hz.
25 26	MS1 MSØ	Mode Select 1 Mode Select Ø	I I	These 2 inputs define the four modes of operation of the CRT 8002 as follows: Alphanumeric Mode — In this mode addresses AØ-A6 (A7 = X) are internally decoded to address 1 of the 128 available ROM characters. The addressed character along with the decoded row will define a 7 bit output from the ROM to be loaded into the shift register via the attribute logic. Thin Graphics Mode — In this mode AØ-A2 (A3-A7 = X) will be loaded into the thin graphic logic along with the row addresses. This logic will define the segments of a graphic entity as defined in figure 2. The top of the entity will begin on row 0000 and will end on a mask programmable row.

MS1	MSØ	MODE
1	1	Alphanumeric
1	0	Thin Graphics
0	1	External Mode
0	0	Wide Graphics

DESCRIPTION OF PIN FUNCTIONS

PIN NO.	SYMBOL	NAME	INPUT/ OUTPUT	FUNCTION
25 26 (cont.)				External Mode – In this mode the inputs AØ-A7 go directly from the character latch into the shift register via the attribute logic. Thus the user may define external character fonts or graphic entities in an external PROM, ROM or RAM. See figure 3. Wide Graphics Mode –In this mode the inputs AØ-A7 will define a graphic entity as described in figure 1. Each line of the graphic entity is determined by the wide graphic logic in conjunction with the row inputs RØ to R3. In this mode each segment of the entity is defined by one of the bits of the 8 bit word. Therefore, the 8 bits can define any 1 of the 256 possible graphic entities. These entities can butt up against each other to form a contiguous pattern or can be interspaced with alphanumeric characters. Each of the entities occupies the space of 1 character block and thus requires 1 byte of memory. These 4 modes can be intermixed on a per character basis.
27	CURSOR	Cursor	I	When this input is enabled 1 of the 4 pre-programmed cursor modes will be activated. The cursor mode is on-chip mask programmable. The standard cursor will be a blinking (at 3.75Hz) reverse video block. The 4 cursor modes are: Underline—In this mode an underline (1 to N raster lines) at the programmed underline position occurs. Blinking Underline—In this mode the underline blinks at the cursor rate. Reverse Video Block—In this mode the Character Block is set to reverse video. Blinking Reverse Video Block—In this mode the Character Block is set to reverse video at the cursor blink rate. The Character Block will alternate between normal video and reverse video. The cursor functions are listed in table 1.
28	RETBL	Retrace Blank	I	When this input is latched high, the shift register parallel inputs are unconditionally cleared to all zeros and loaded into the shift register on the next Load/Shift pulse. This blanks the video, independent of all attributes, during horizontal and vertical retrace time.

TABLE 1

CURSOR	RETBL	REVID	CHABL	UNDLN*	FUNCTION
X	1	X	X	X	"0" (S.R. All
0	0	0	0	0	D (S.R.) All
0	0	0	0	1	"1" (S.R.)*
					D (S.R.) All others
0	0	0	1	X	"0" (S.R.) All
0	0	1	0	0	D (S.R.) All
0	0	1	0	1	"0" (S.R.)*
					D (S.R.) All others
0	0	1	1	X	"1" (S.R.) All
Underline*	0	0	0	X	"1" (S.R.)*
					D (S.R.) All others
Underline*	0	0	1	X	"1" (S.R.)*
					"0" (S.R.) All others
Underline*	0	1	0	X	"0" (S.R.)*
					D (S.R.) All others
Underline*	0	1	1	X	"0" (S.R.)*
					"1" (S.R.) All others
Blinking** Underline*	0	0	0	X	"1" (S.R.)* Blinking
					D (S.R.) All others
Blinking** Underline*	0	0	1	X	"1" (S.R.)* Blinking
					"0" (S.R.) All others
Blinking** Underline*	0	1	0	X	"0" (S.R.)* Blinking
					D (S.R.) All others
Blinking** Underline*	0	1	1	X	"0" (S.R.)* Blinking
					"1" (S.R.) All others
REVID Block	0	0	0	0	D (S.R.) All
REVID Block	0	0	0	1	"0" (S.R.)*
					D (S.R.) All others
REVID Block	0	0	1	X	"1" (S.R.) All
REVID Block	0	0	0	1	"0" (S.R.)*
					D (S.R.) All others
REVID Block	0	1	0	0	D (S.R.) All
REVID Block	0	1	0	1	"1" (S.R.)*
					D (S.R.) All others
REVID Block	0	1	1	X	"0" (S.R.) All
Blink** REVID Block	0	0	0	0	Alternate Normal Video/REVID At Cursor Blink Rate
Blink** REVID Block	0	0	0	1	
Blink** REVID Block	0	0	1	X	
Blink** REVID Block	0	1	0	0	
Blink** REVID Block	0	1	0	1	
Blink** REVID Block	0	1	1	X	

*At Selected Row Decode **At Cursor Blink Rate

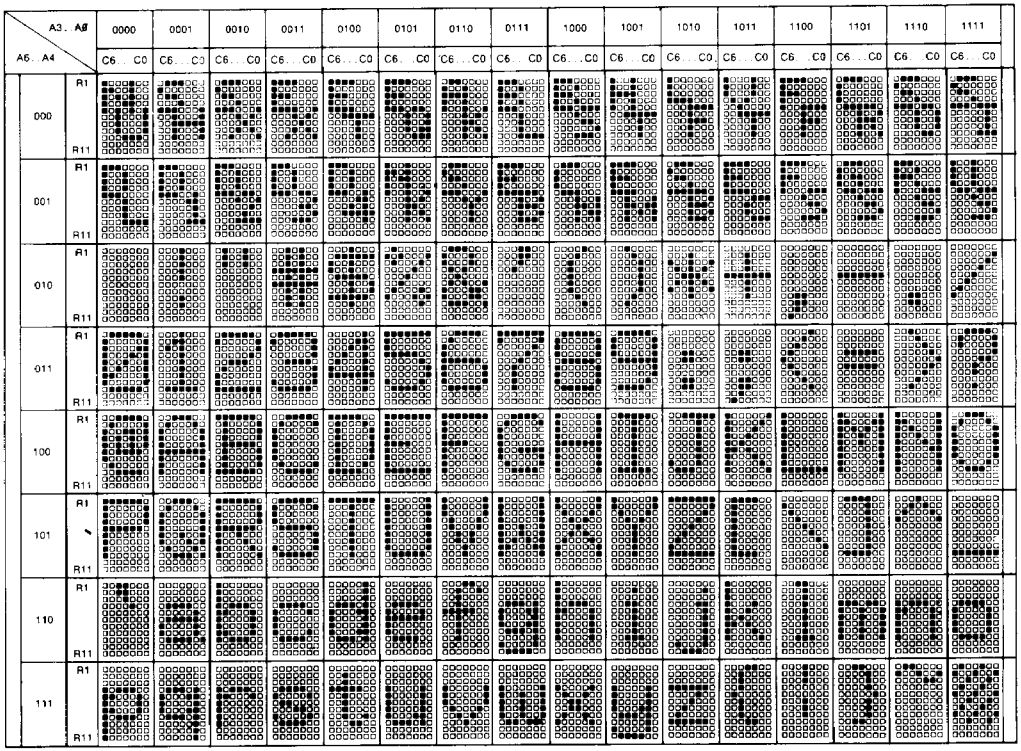
Note: If Character is Blinking at Character Rate, Cursor will change it to Cursor Blink Rate.

FIGURE 5
ROM CHARACTER BLOCK FORMAT

											ROWS	R3	R2	R1	RØ
(ALL ZEROS) →	0	0	0	0	0	0	0	0	—	—	RØ	0	0	0	0
	0	0	0	0	0	0	0	0	—	—	R1	0	0	0	1
	0	0	0	0	0	0	0	0	—	—	R2	0	0	1	0
	0	0	0	0	0	0	0	0	—	—	R3	0	0	1	1
	0	0	0	0	0	0	0	0	—	—	R4	0	1	0	0
	0	0	0	0	0	0	0	0	—	—	R5	0	1	0	1
77 BITS (7 x 11 ROM)	0	0	0	0	0	0	0	0	—	—	R6	0	1	1	0
	0	0	0	0	0	0	0	0	—	—	R7	0	1	1	1
	0	0	0	0	0	0	0	0	—	—	R8	1	0	0	0
	0	0	0	0	0	0	0	0	—	—	R9	1	0	0	1
	0	0	0	0	0	0	0	0	—	—	RIØ	1	0	1	0
	0	0	0	0	0	0	0	0	—	—	R11	1	0	1	1
	0	0	0	0	0	0	0	0	—	—	R12	1	1	0	0
	0	0	0	0	0	0	0	0	—	—	R13	1	1	0	1
(ALL ZEROS)	0	0	0	0	0	0	0	0	—	—	R14	1	1	1	0
	0	0	0	0	0	0	0	0	—	—	R15	1	1	1	1
	*C7	C6	C5	C4	C3	C2	C1	CØ	—	—					

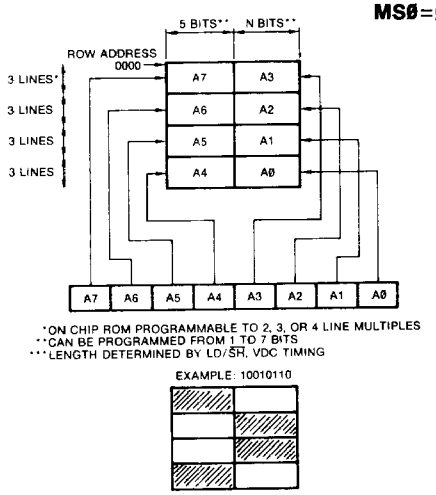
EXTENDED ZEROS (BACK FILL) FOR INTERCHARACTER SPACING (NUMBER CONTROLLED BY LD/SH, VDC TIMING)

*COLUMN 7 IS ALL ZEROS (REVID = 0)
COLUMN 7 IS SHIFTED OUT FIRST



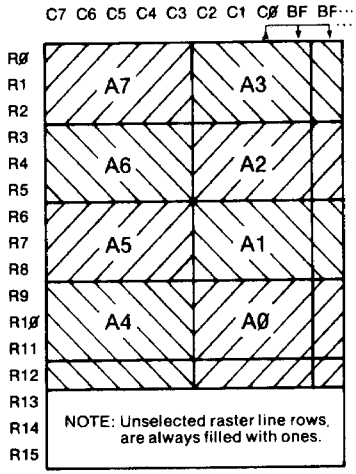
CONSULT FACTORY FOR CUSTOM FONT AND OPTION PROGRAMMING FORMS.

FIGURE 1
WIDE GRAPHICS MODE
MSØ=Ø MS1=Ø



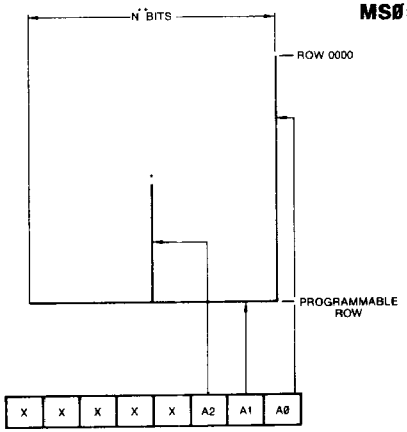
*ON CHIP ROM PROGRAMMABLE TO 2, 3, OR 4 LINE MULTIPLES
**CAN BE PROGRAMMED FROM 1 TO 7 BITS
***LENGTH DETERMINED BY LD/$\overline{SR}$, VDC TIMING

EXAMPLE: 10010110

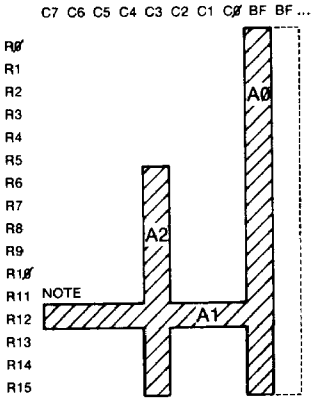


BF = back fill

FIGURE 2
THIN GRAPHICS MODE
MSØ=Ø MS1=1



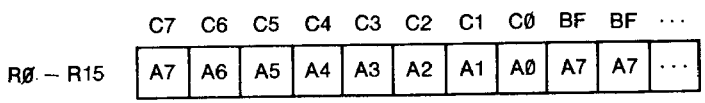
X = DON'T CARE
* THE INSIDE SEGMENT IS MASK PROGRAMMABLE TO ROW 0000
** LENGTH DETERMINED BY LD/$\overline{SR}$, VDC TIMING



NOTE: When A1 = "1", the underline row/rows are deleted.
When A1 = "0", the underline, if selected, will appear.

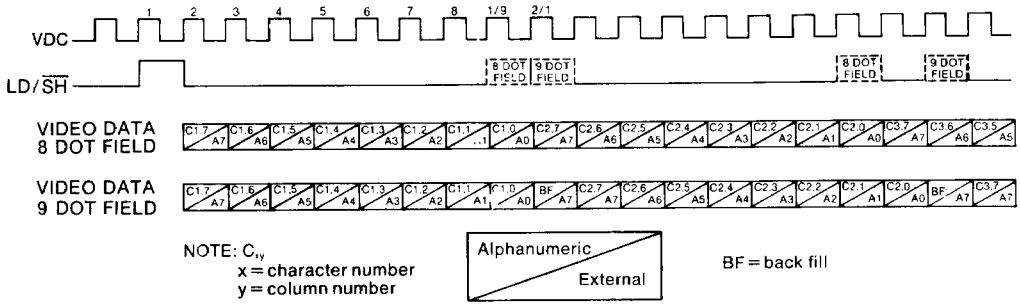
BF = back fill

FIGURE 3
EXTERNAL MODE
MSØ=1 MS1=Ø



BF = back fill

FIGURE 4 TYPICAL VIDEO OUTPUT



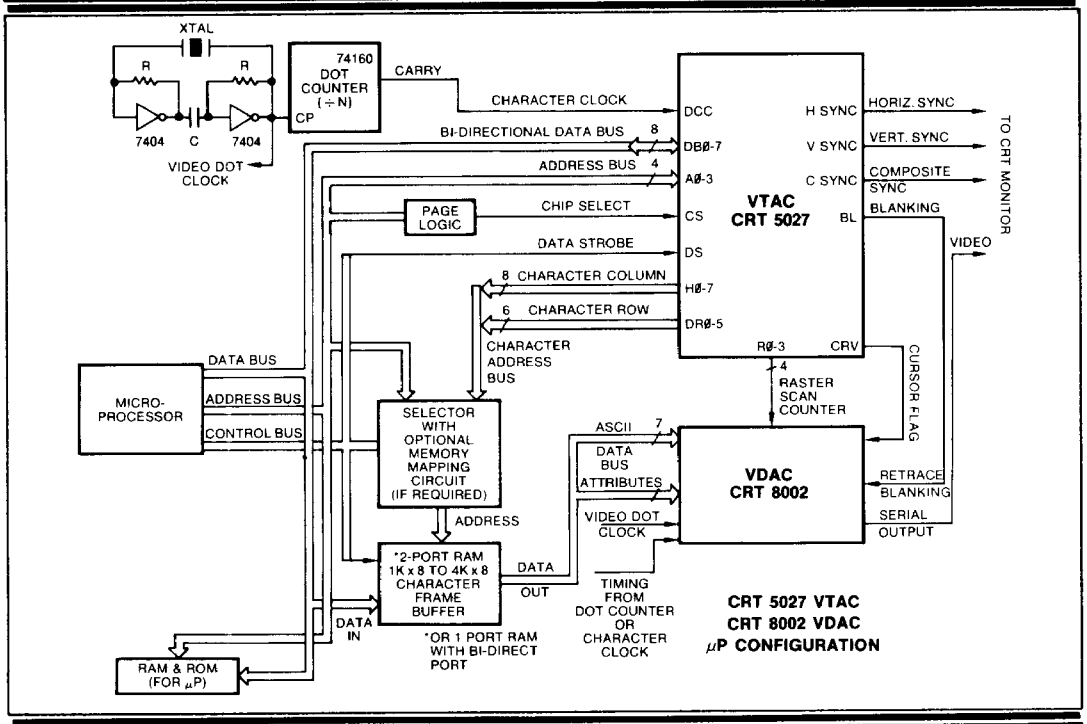
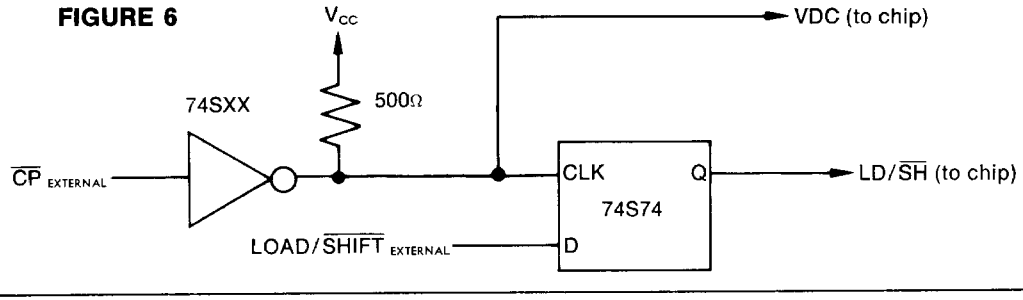
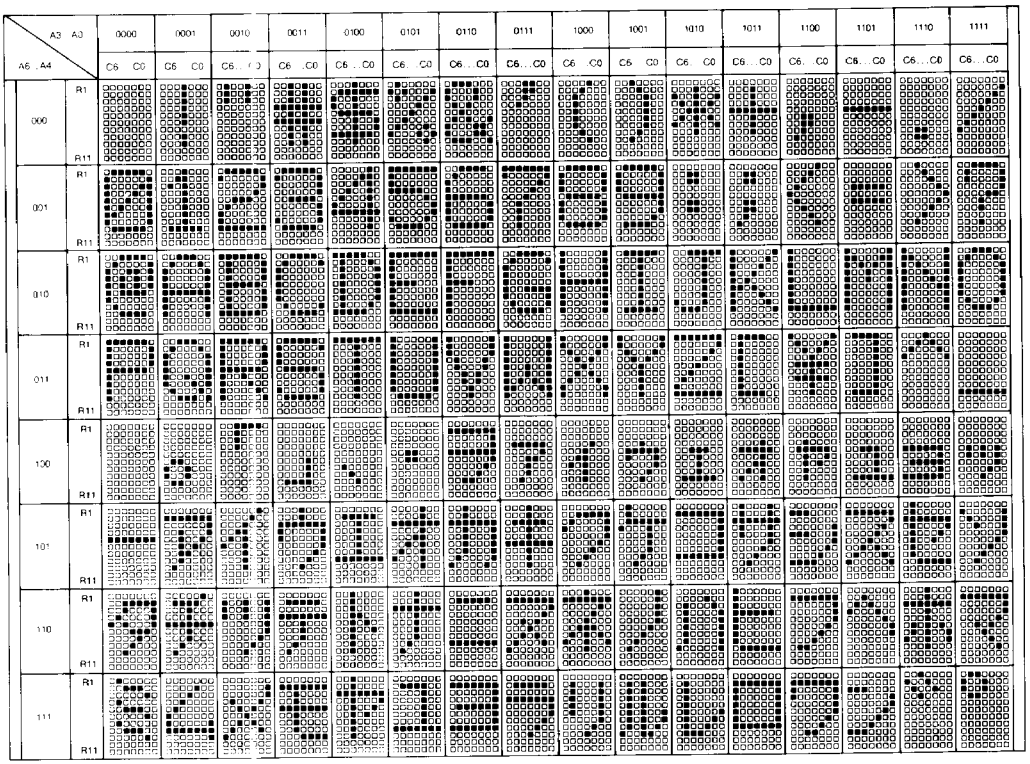


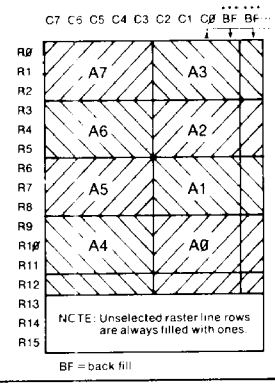
FIGURE 6



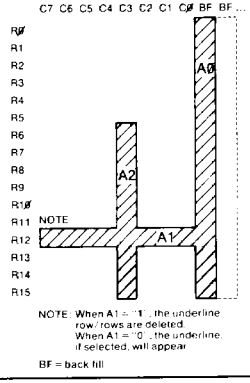
CRT Video Display-Controller Video Generator VDAC™



WIDE GRAPHICS MODE



THIN GRAPHICS MODE



ATTRIBUTES

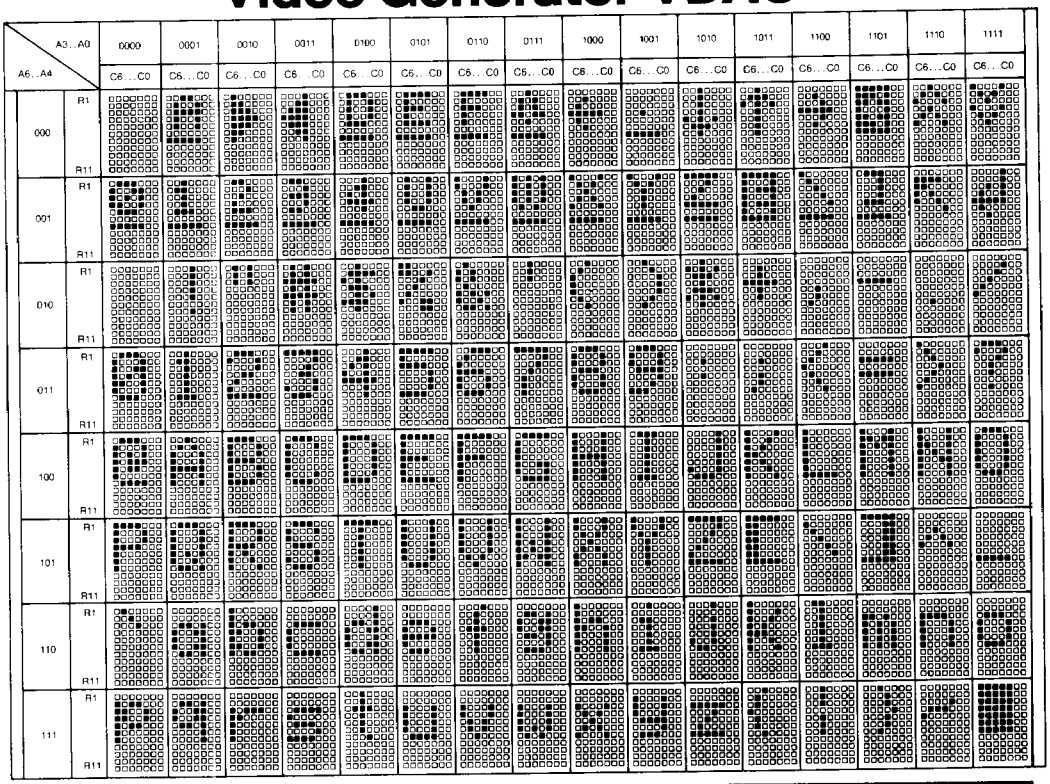
Underline
Underline will be a single horizontal line at row R11

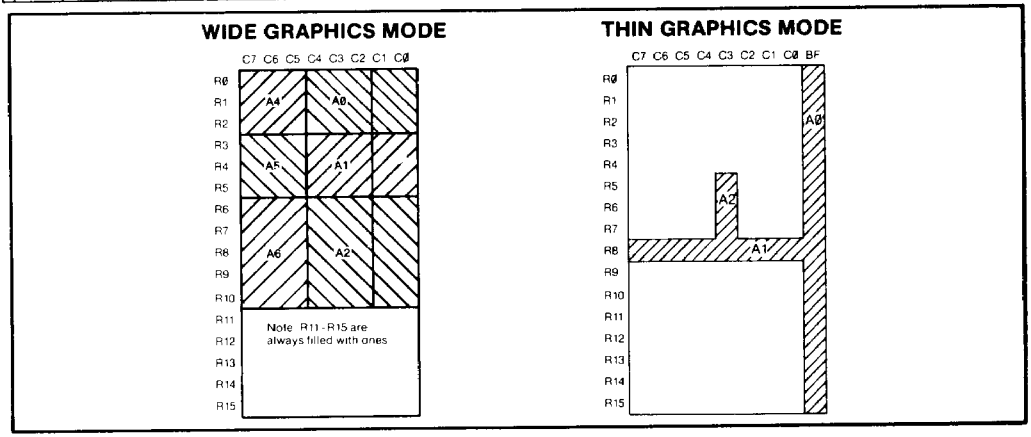
Cursor
Cursor will be a blinking reverse video block, blinking at 3.75 Hz

Blink Rate
The character blink rate will be 1.875 Hz

Strike-Thru
The strike-thru will be a double line at rows R5 and R6

CRT Video Display-Controller Video Generator VDAC™





ATTRIBUTES

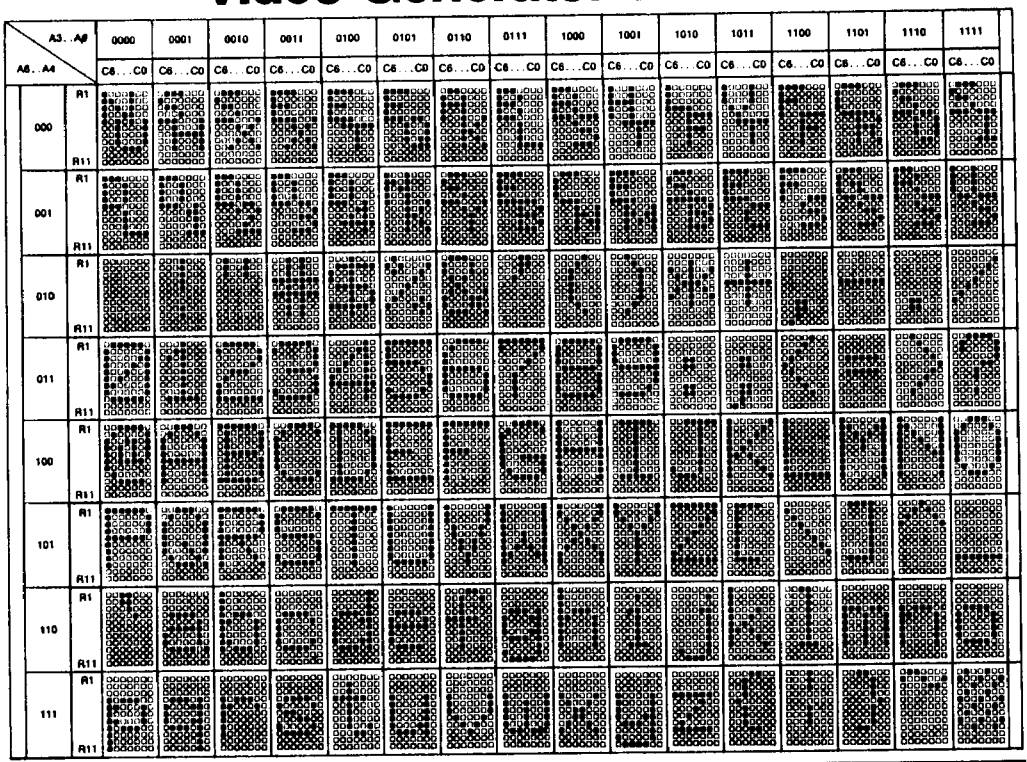
Underline
Underline will be a single horizontal line at R8

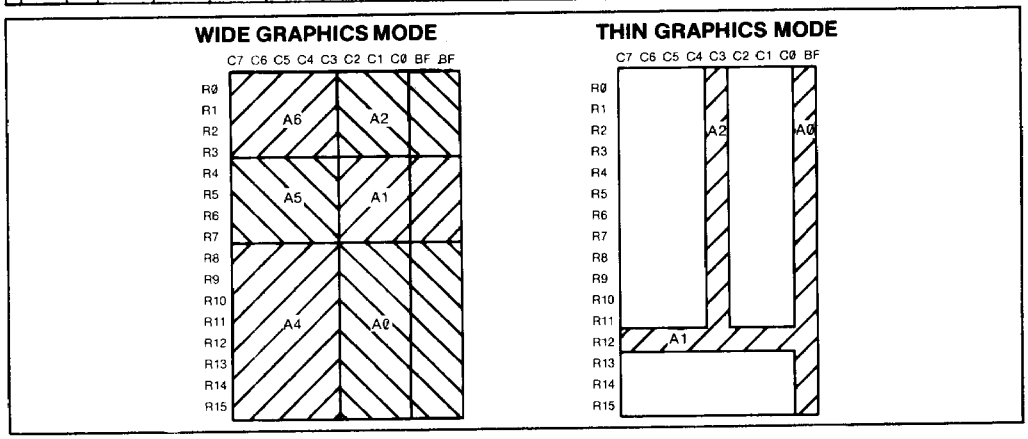
Cursor
Cursor will be a blinking reverse video block, blinking at 3.75 Hz

Blink Rate
The character blink rate is 1.875 Hz

Strike-Thru
The strike-thru will be a single horizontal line at R4

CRT Video Display-Controller Video Generator VDAC™





ATTRIBUTES

Underline
Underline will be a single horizontal line at R12

Cursor
Cursor will be a reverse video block

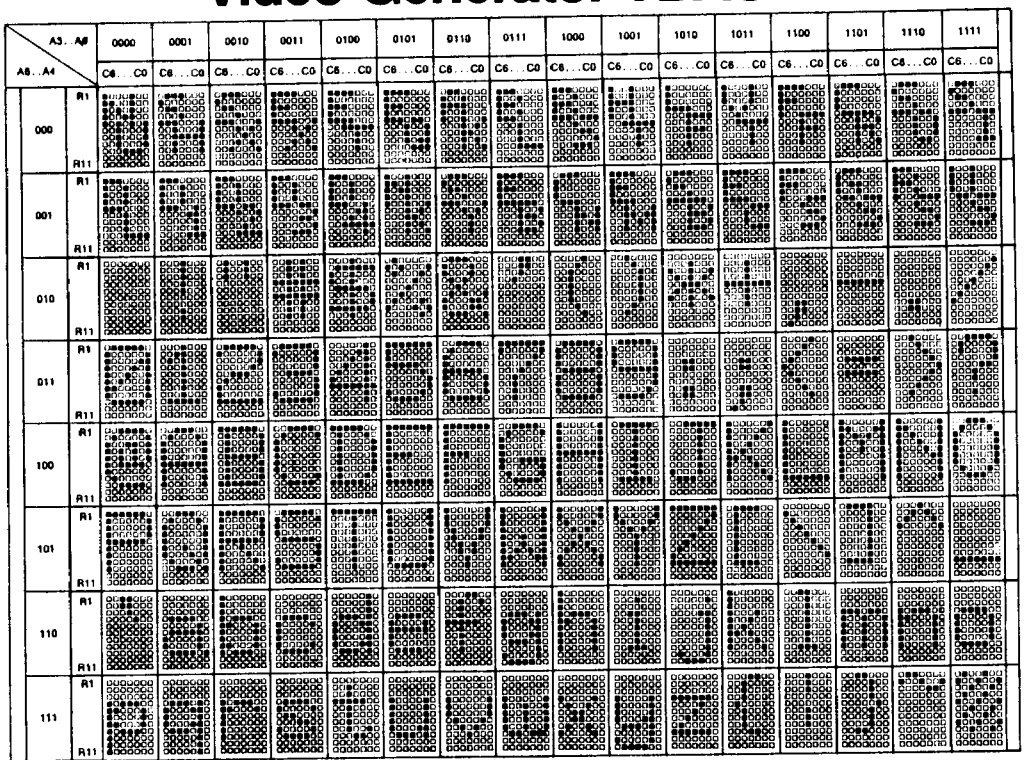
Blink Rate
The character blink rate is 1.875 Hz

Strike-Thru
The strike-thru will be a double line at rows R5 and R6

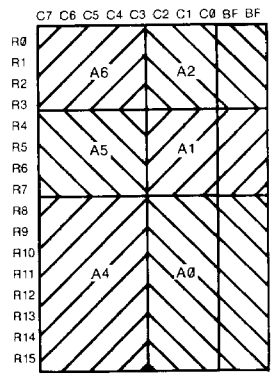
STANDARD MICROSYSTEMS CORPORATION

CRT 8002-011 (ASCII) CODING INFORMATION

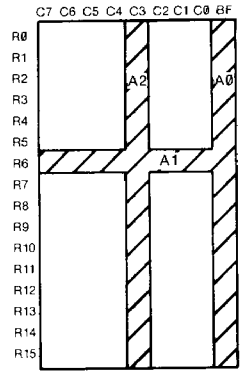
CRT Video Display-Controller Video Generator VDAC™



WIDE GRAPHICS MODE



THIN GRAPHICS MODE



ATTRIBUTES

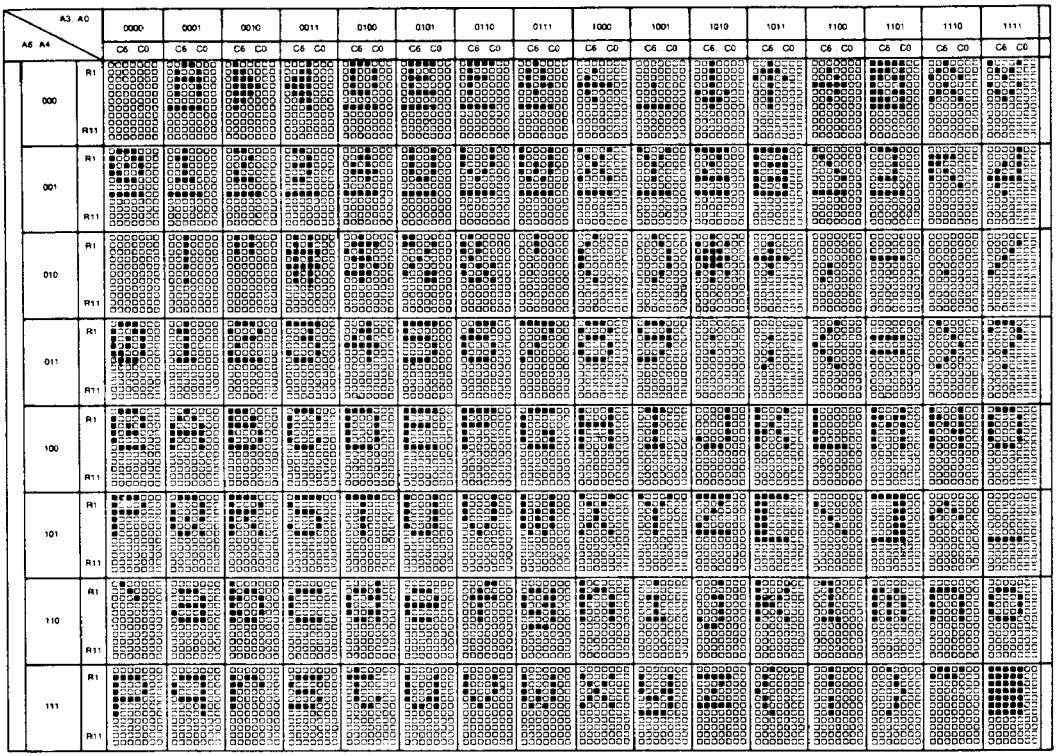
Underline
Underline will be a single horizontal line at R11

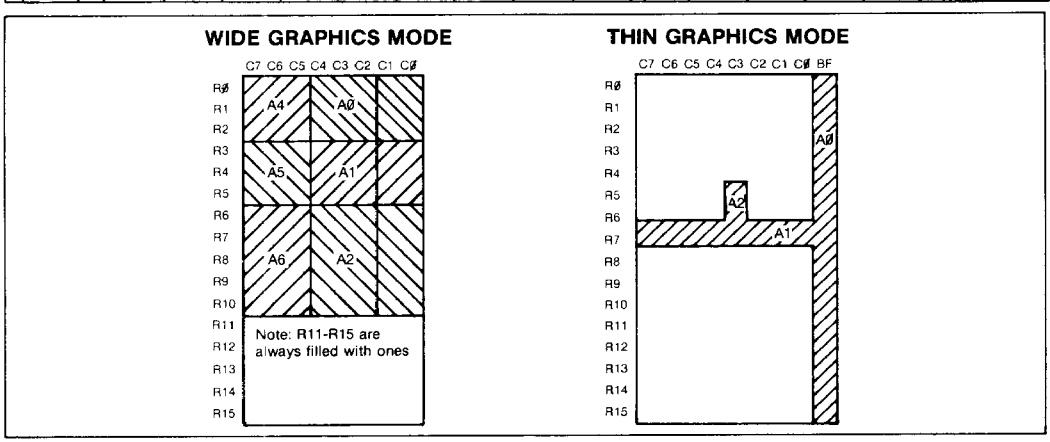
Cursor
Cursor will be a blinking reverse video block, blinking at 3.75 Hz

Blink Rate
The character blink rate is 1.875 Hz

Strike-Thru
The strike-thru will be a double line at rows R5 and R6

CRT Video Display-Controller Video Generator VDAC™





ATTRIBUTES

Underline
Underline will be a double horizontal line at R7 and R8

Cursor
Cursor will be a reverse video block

Blink Rate
The character blink rate is 1.875 Hz

Strike-Thru
The strike-thru will be a single horizontal line at R4

STANDARD MICROSYSTEMS CORPORATION
35 Marcus Blvd., Hauppauge, N.Y. 11788
(516) 273-3100 TWX-510-227-8898