

VIDEO TERMINAL INTERFACE

MANUAL

Copyright 1976 IPC

POLYMORPHIC SYSTEMS
737 South Kellogg Avenue
Goleta, California 93017

PolyMorphic Systems Video Terminal Interface
Assembly and Operation

	Table of Contents	Page
1.	Assembly Instructions	1-1
1.1	Check components	1-1
1.2	Install integrated circuits	1-5
1.3	Install discrete components	1-8
1.4	Select line length	1-10
1.5	Select address location	1-11
1.6	Interface TV monitor or receiver	1-12
1.7	Connect keyboard	1-13
1.7.1	connector configuration	1-14
1.7.2	keypress strobe	1-15
1.7.3	optional voltage regulator	1-17
1.8	Install optional on-board clock	1-17
2.	Theory of Operation	2-1
2.1	Block diagram	2-1
2.2	Schematic diagram	2-3
2.2.1	character generation	2-4
2.2.2	raster and timing	2-5
2.2.3	character and raster synchronization	2-7
2.2.4	external bus and keyboard interfacing	2-9
3.	Troubleshooting	3-1
3.1	Power mains	3-2
3.2	Signal tracing	3-3
3.2.1	video interface	3-5

Table of Contents, Continued

	Page
3.2.2 localizing on the video path	3-5
3.2.3 localizing on the EOC (end of character) path	3-8
3.2.4 localizing on the dot clock path	3-9
3.2.5 localizing on the horizontal blanking path	3-10
3.2.6 sweep and symbol relates counter patterns	3-12
3.3 Diagnostic aids	3-15
3.3.1 time calibration	3-16
4. Software	4-1
4.1 Video terminal software	4-1
4.2 Graphics software	4-8

Appendixes

- A. ASCII Character Set
- B. Character Fonts
- C. Chip Pinouts
- D. Graphics Character Set

Introducing the PolyMorphic Systems Video Terminal Interface

Here is your PolyMorphic Systems video terminal interface (VTI). With it, you can display 16 lines of 32 or 64 characters (dependent on options) on a TV monitor or slightly modified TV receiver. The character-forming dot matrix projected onto the TV screen is seven dots wide by nine dots high, embedded in a ten-by-fifteen field, for distinct, natural-looking, easily readable symbols. The VTI will display upper and lower case letters, punctuation, numbers, a special set of symbols representing the ASCII control characters, plus another special set of 64 graphics plotting characters. The special graphics capability allows you to display a plotting grid with up to 48 by 128 cells in addition to any other characters. Board circuitry includes an input port for an ASCII keyboard, making it a complete interface between keyboard, computer, and TV. If you have purchased an assembled unit, proceed to section 1.5.

1. Assembly Instructions

1.1 Component check

The video terminal interface kit contains several bags of parts. The standard kit, which produces lines of 32 characters on the video screen, consists of 3 bags of parts, number 0 through 2. The memory option (for a 64-character line), if ordered, is in the bag numbered 3. A complete socket set, also optional, is in the bag numbered 4. Check the kit to be sure you received

PolyMorphic Systems

Video Rev 0.0

Page 1-2

the correct bags. Check each bag according to the parts list on the following page to see that the parts are all present. If the parts or their equivalents are not provided as listed, see the warranty information sheet.

VIDEO TERMINAL INTERFACE PARTS LISTBag #0 Integrated circuits

Qty. P/N

4 AM9111A or P2111A-1 or AM91L11A
1 MCM6571A or MCM6574 or MCM6576
1 8212 or SN74S412 or MCM 6575
1 N8274
1 DM8131
2 74LS00
1 74LS02
1 7407
1 74LS20
2 74LS74
1 74123 or 74LS123
2 74LS138
2 74150 or 74LS150
5 74LS157
3 74161 or 74LS161
3 DM8097 or N8T97 or SN74367
1 74273 or 74LS273
2 SN74393
1 74S124
1 24 pin socket (for IC #36)
1 14 pin socket (for keyboard plug)
1 74LS132

BAG #1

1✓ 82 ohm $\frac{1}{4}$ W 10% (grey-brown-black)
1 150 ohm $\frac{1}{2}$ W 10% (brown-green-brown)
2✓ 220 ohm $\frac{1}{4}$ W 10% (red-red-brown)
1✓ 470 ohm $\frac{1}{4}$ W 10% (yellow-violet-brown)
2✓ 10K ohm trimpot
1 1N5225B 3.0 V Zener
1 1N759 12.0 V Zener
1 LM340T 5.0 or MC7805P
3✓ 1000 ohm $\frac{1}{4}$ W 10% (brown-black-red)
2✓ 4700 ohm $\frac{1}{4}$ W 10% (yellow-violet-red)
1 0.0047 F + 10% CW 15 472K
1✓ 2N5449 transistor
1 27 PF capacitor
1 22 PF capacitor
1 39 PF capacitor
5 10 F capacitor

VIDEO TERMINAL INTERFACE PARTS LISTBAG #2

Qty. P/N

10	2200 ohm $\frac{1}{4}$ W 10% (red-red-red)
21	.1 F capacitor
1	6106B-14 heat sink
1	6-32 X 3/8 machine screw
1	#6 hex nut
1	#6 lockwasher

solder, wire, sleeving

BAG #3 Memory option

4	AM91L11A
4	.1 F capacitor

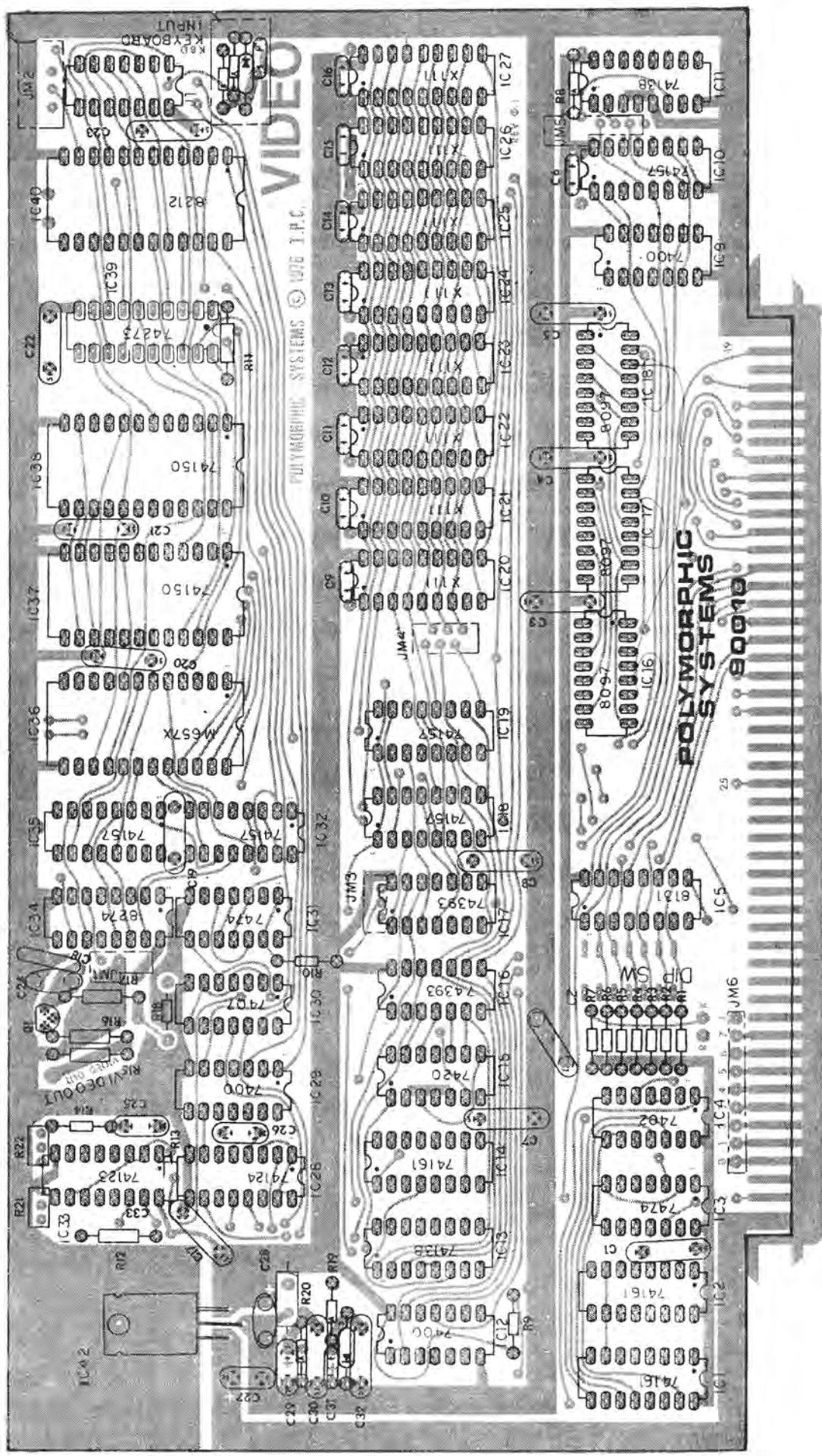
BAG #4 Socket kit

3	24 pin socket
1	20 pin socket
8	18 pin socket
17	16 pin socket
10	14 pin socket
1	14 pin dip switch

1 circuit board

1.2 Install integrated circuits

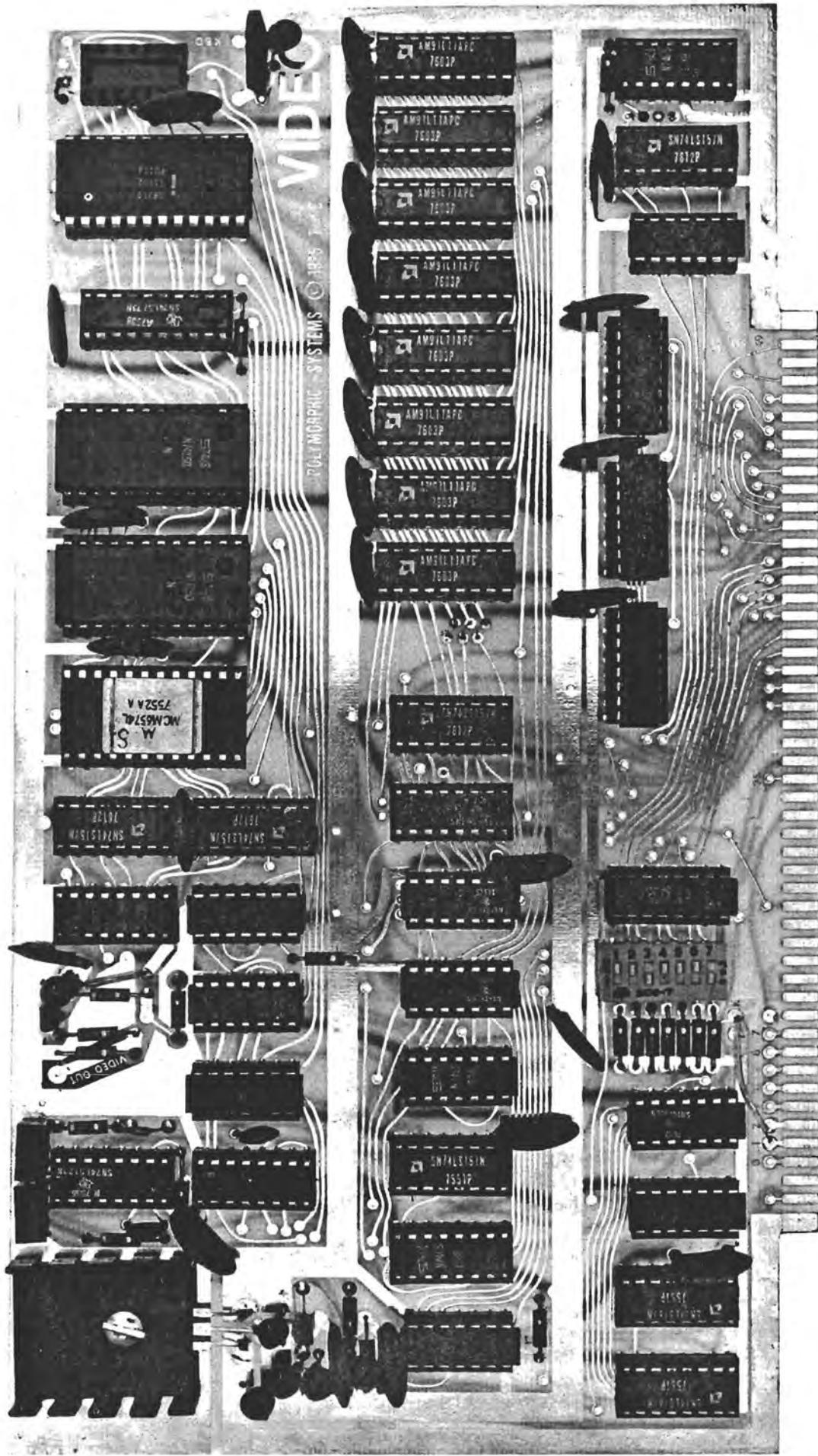
Load all components from the TOP of the board, which is viewed when the 100 pin bus connector is pointing down and the large etched word "VIDEO" is legible at the upper right. If you have purchased the socket option, insert the appropriate sockets as indicated by the column giving the number of pins on each device. Insert the dual-in-line packages into the board by referring to the order of installation on page and to figure 1-1 for each device location and orientation. For the 32 character line, IC's 22, 23, 24, and 25 are omitted, as these chips comprise the memory option. Solder all pins on each device from the bottom of the board, being careful not to create solder bridges to adjacent pins or traces. If you are not familiar with DIP insertion and soldering techniques, refer to the enclosed sheet on DIP handling.



<u>CHECK</u>	<u>IC #</u>	<u>TYPE</u>	<u>#PINS</u>	<u>FUNCTION</u>
{ J)	1	74161	16	Binary counter
{)	2	74161	16	Binary counter
{)	3	74LS74	14	Dual D flip-flop
{)	4	74LS02	14	Quad 2-input NOR gate
{)	5	8131	16	6 input comparator
{)	6	8097	16	Hex tri-state buffer
{)	7	8097	16	Hex tri-state buffer
{)	8	8097	16	Hex tri-state buffer
{)	9	74LS00	14	Quad 2-input NAND gate
{)	10	74LS157	16	Quad 2-input multiplexor
{)	11	74LS138	16	1 of 8 decoder
{)	12	74LS132	14	Quad 2-input NAND gate
{)	13	74LS138	16	1 of 8 decoder
{)	14	74161	16	Binary counter
{)	15	74LS20	14	Dual 4-input NAND gate
{)	16	74393	14	Dual 4 bit binary counter
{)	17	74393	14	Dual 4 bit binary counter
{)	18	74LS157	16	Quad 2-input multiplexor
Omit {)	19	74LS157	16	Quad 2-input multiplexor
22-25 {)	20-27	2111-1 or 9111	18	256X4 bit static memory
for 32 {)	28	74S124	16	Dual gated voltage contrl'd osc.
char. {)	29	74LS00	14	Quad 2-input NAND gate
{)	30	7407	14	Hex open-collector buffer
{)	31	74LS74	14	Dual D flip-flop
{)	32	74LS157	16	Quad 2-input multiplexor
{)	33	74123	16	Dual monostable multivibrator
{)	34	8274	16	Ten bit shift register
{)	35	74LS157	16	Quad 2-input multiplexor
{)	36	24 pin socket	24	character generator ROM socket
{)	37-38	74150	24	16-input multiplexor
{)	39	74273*	20	8 bit register
{)	40	8212	24	8 bit latch
{)	41	14 pin socket	14	Keyboard input connector

() The last integrated circuit, IC42, is a 7805 or LM340-5.0 volt regulator. It is not in a dual-in-line package but has a metal tab and three leads. Place the hole in the metal tab over the hole in the large heat sink area on the upper left of the circuit board. Orient the three leads downward over the three

* The polarity may be ambiguous; the oblong groove indicates the device orientation.



holes shown in figure 1-1, and note where to bend them to go through. Bend them with small pliers, and check to see that when the three leads go through the board the mounting holes line up. Then insert the 6-32 screw from the bottom of the board, place the heat sink over the screw from the top, and insert the regulator leads into the board while the tab slips over the machine screw. Use the lockwasher and nut to secure the regulator and heat sink to the board. Solder the three leads and trim them.

1.3 Install discrete components

- (✓) Insert .1 F ceramic disc bypass capacitors C1 through C10 and C15 through C23 as shown in figure 1-1. If you have purchased the 64 character option, insert capacitors C11 through C14 adjacent to the optional memory chips.
- (✓) Insert 2200 $\frac{1}{2}$ W resistors (red-red-red) R1 through R6, R8, R10, R11, and R13.
- (✓) Insert 1K $\frac{1}{4}$ W resistors (brown-black-red) R7, R9, and R19.
- (✓) Insert 4.7 K $\frac{1}{4}$ W resistors (yellow-violet-red) R12 and R14.
- (✓) Insert 82 $\frac{1}{4}$ W (gray-red-black) resistor R15.
- (✓) Insert 220 $\frac{1}{4}$ W (red-red-brown) resistors R16 and R18.*
- (✓) Insert 470 $\frac{1}{4}$ W (yellow-violet-brown) resistor R 17.
- () Insert 150 $\frac{1}{2}$ Watt resistor R20. This may need to be installed vertically over the right-hand pad.
- (✓) Insert 3 volt zener diode Z1 as shown in figure 1-1. Note the polarity of the diode as indicated on the board. (1N5255)

* For details on how to install resistor R18, see next page.

- () Insert 12 volt zener diode Z2 as shown. Note the polarity as indicated on the board. (1N759)
- (✓) Insert 10 F tantalum capacitors C24, C27, C28, C29 and C31. Note the polarity marked on the board and indicated by a + or dot on each capacitor. C24 is oriented with the positive side nearest the edge connector.
- (✓) Insert .1 F ceramic disc capacitors C30 and C32 as shown in figure 1-1.
- (✗) Insert 4700 pF capacitor C25.
- (✓) Insert 27 pF capacitor C33
- () Insert 10K trim potentiometers R21 and R22.
- (✓) Cut a piece of teflon tubing about 5/8" long. Insert jumper in area adjacent to IC34 as shown in Fig. 1-1
- (✗) Insert 2N5449 transistor Q1 as shown. C E
B Bottom view
- (✗) Insert capacitor C26. If you have the 32 character option the value is 39 PF. If you have the 64 character option the value is 22 PF.

Solder all components on the back of the board and trim the leads.

Resistor R18 is installed at a point in the circuitry that is also convenient to use as a troubleshooting test point, as described in Section 4. Troubleshooting requires disconnecting the right side of R18. To make disconnection easy, insert a bit of trimmed-off component lead into the right-side mounting hole of R18 (immediately adjacent to pin 7 of IC 30) and solder it to the board. Then attach the right lead of R18 to this bit of wire.

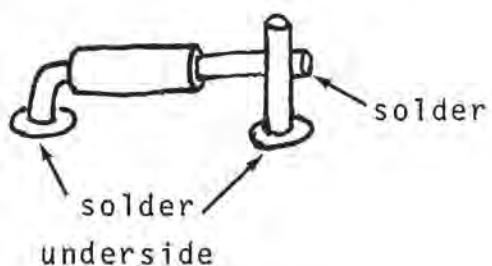


Figure 1-2

1.4 Select line length

Several jumper pads on the board must be connected to allow the memory addresses to be scanned in 32-character lines or 64-character lines. Near the center of the board is a region resembling that depicted in figure 1-3. On the lower right is a region similar to figure 1-4.

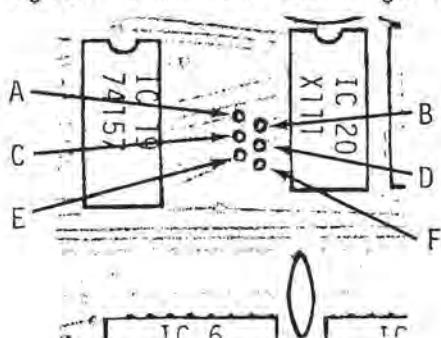


Figure 1-3

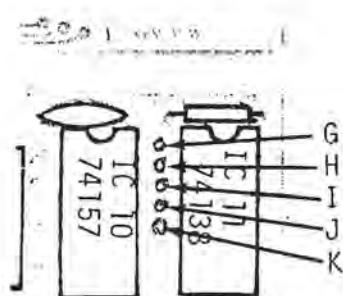


Figure 1-4

Make connections between the pads as shown in the following table. Use short lengths of the uninsulated wire provided, except for the long connection between F and G.

32-character line

A no connection
B to C
D to E
F to G (use sleeved wire provided)
H to I
J to K

64-character line

A to B
C to D
E to F
G to H
I to J
K no connection

In addition, there are three jumper pads at the top of IC17 (adjacent to pins 1 and 14). For a 32-character line, jumper from the center pad to the pad nearest pin 1. To get a 64-character line, jumper from the center pad to the pad nearest pin 14 of IC 17.

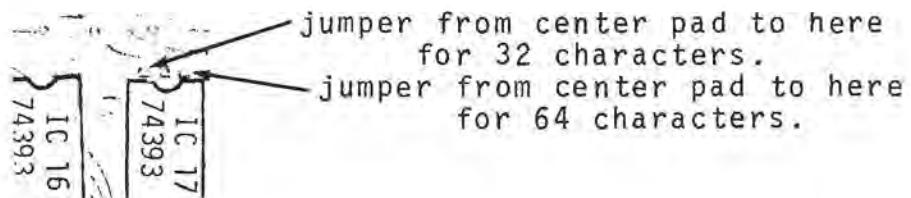


Figure 1-5

1.5 Select address location

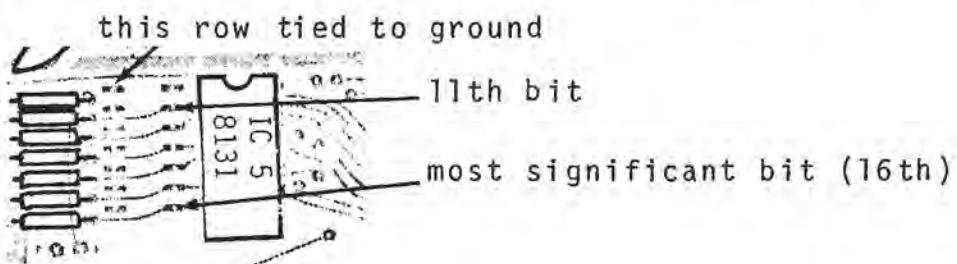
The video terminal interface interacts with the Altair bus as a block of memory and an input port. The block of memory ($\frac{1}{2}$ K or 1K bytes, depending on the line length option) can be located at any address from 0 to 63K in 1K increments*. Location is determined by comparing the 6 most significant bits of the memory address with 6 jumper selected bits. To reduce the complexity of the circuitry, the memory block selection also determines the address of the input port. The 6 most significant bits of

* Software written for this product will generally locate it at hexadecimal address 8800.

the one byte address for the input port must match the six jumper selected bits. The two least significant bits are not compared for the input address, so their state is arbitrary.

The address selection jumper area is located in the lower left hand region of the board, adjacent to IC5. Each of the six most significant bits of the address is tied with a resistor to +5V, so that they are normally in the binary state 1. Any or all of them may be jumpered to ground to put them in the binary state 0. Note the two rows of pads to the left of IC5 (see Fig. 106). These pads are a dual-in-line spacing so that a DIP switch may be used for address selection, if desired. Normally the address line on the right may be jumpered or switched directly to the grounded pads on the left, to produce a logical zero, with the bit sequence as shown in Figure 1-6.

Figure 1-6



1.6 Interface TV monitor or receiver

As it stands now, your unit should work if connected via coaxial cable to a video monitor or modified television set.* Because of rigid FCC requirements, the circuit has been designed to be

* If random characters do not appear on "power up," some adjustment of the horizontal frequency on the back of the receiver or monitor may be required.

connected directly to the video input circuit of a television set. The video board's output must be connected to the input of the television's video amplifier, which is located between the last video IF stage and the video output circuit. When you break the circuit right at the input to the video amplifier, you will probably have to provide a DC bias circuit for the stage, since in most cases it is supplied by the now-disconnected video IF amplifier. The best interface circuit will vary from set to set, but frequently a capacitive coupling to a resistive bias circuit is adequate. The coupling capacitor is typically a 1-5 μ f tantalum, oriented with the positive side connected to the video input amplifier. **IMPORTANT:** Check to see that the chassis of your TV is isolated by a transformer from the 110 VAC line. If the chassis is not so isolated, but rather a polarized plug has been used on the line cord, FATAL INJURY COULD RESULT from possible electrical shock. If you must use this type of set, either isolate it with a transformer or isolate the video signal with an opto-isolator between the video terminal interface and the video input connection to the TV set. Under no circumstances should the polarized plug be trusted to maintain the isolation from the line voltage.

1.7 Connect keyboard

At the upper right hand corner of the video terminal interface board is the keyboard input port. This port provides a latched 8 bit parallel input capability which completely interfaces with

many ASCII keyboards. Keyboards usually indicate a keystrike to the computer via a strobe line, in addition to the eight parallel input lines. The signal on this line changes state -- from high to low or from low to high -- to indicate a keystrike. Hookup varies according to whether the stroke on your keyboard is "positive going" (rising in voltage to indicate keystrike) or "negative going" (dropping to indicate keystrike).

1.7.1 Connector configuration

The parallel input from the keyboard is designed to come in over a ribbon cable terminated by a dual-in-line header. This header plugs into the 14 pin DIP socket at the extreme upper right hand corner of the board. The 8 parallel input lines are connected to pins 1 through 8 of this socket, with 1 being the least significant bit. Pin 9 carries the strobe, which may be "positive going" or "negative going," as discussed in the next section. Pins 10, 11, and 12 are grounded. Pin 13 is the output from the optional* negative voltage regulator, as discussed in section 2.3.3. Pin 14 carries +5 volts as the primary supply for most keyboards.

Figure 1.7
Keyboard Input
Connector

DATA IN 0	1	14	+5 VOLTS
DATA IN 1	2	13	NEG REG
DATA IN 2	3	12	
DATA IN 3	4	11	GND
DATA IN 4	5	10	
DATA IN 5	6	9	KEY STROBE
DATA IN 6	7	8	DATA IN 7

* Used when the keyboard requires a negative supply. The user should select and obtain the components suited to his keyboard.

1.7.2 keypress strobe

When the processor accesses the video terminal interface with an input instruction, the state of the keyboard input latch is transferred to the accumulator. However, proper use of the keyboard requires that the processor establish two conditions before using the input data. It must establish that

- 1) a key has been pressed

and

- 2) this particular key depression has not been previously serviced.

These functions are accomplished by making the keypress strobe information available to the processor.

The keypress strobe line is an additional keyboard output line running in parallel with the data lines. This line signals every key depression by a pulse. This pulse can be used in two ways to inform the processor that the necessary input conditions are met. The pulse can be used to

- 1) interrupt the processor by setting an interrupt service latch contained on the input buffer,
or
- 2) indicate the keyboard state via the most significant data bit. This is possible since the ASCII character set can be encoded in 7 bits.

The jumper connections which allow the strobe line to be used on an interrupt or a status bit basis are shown in figure 1-8.

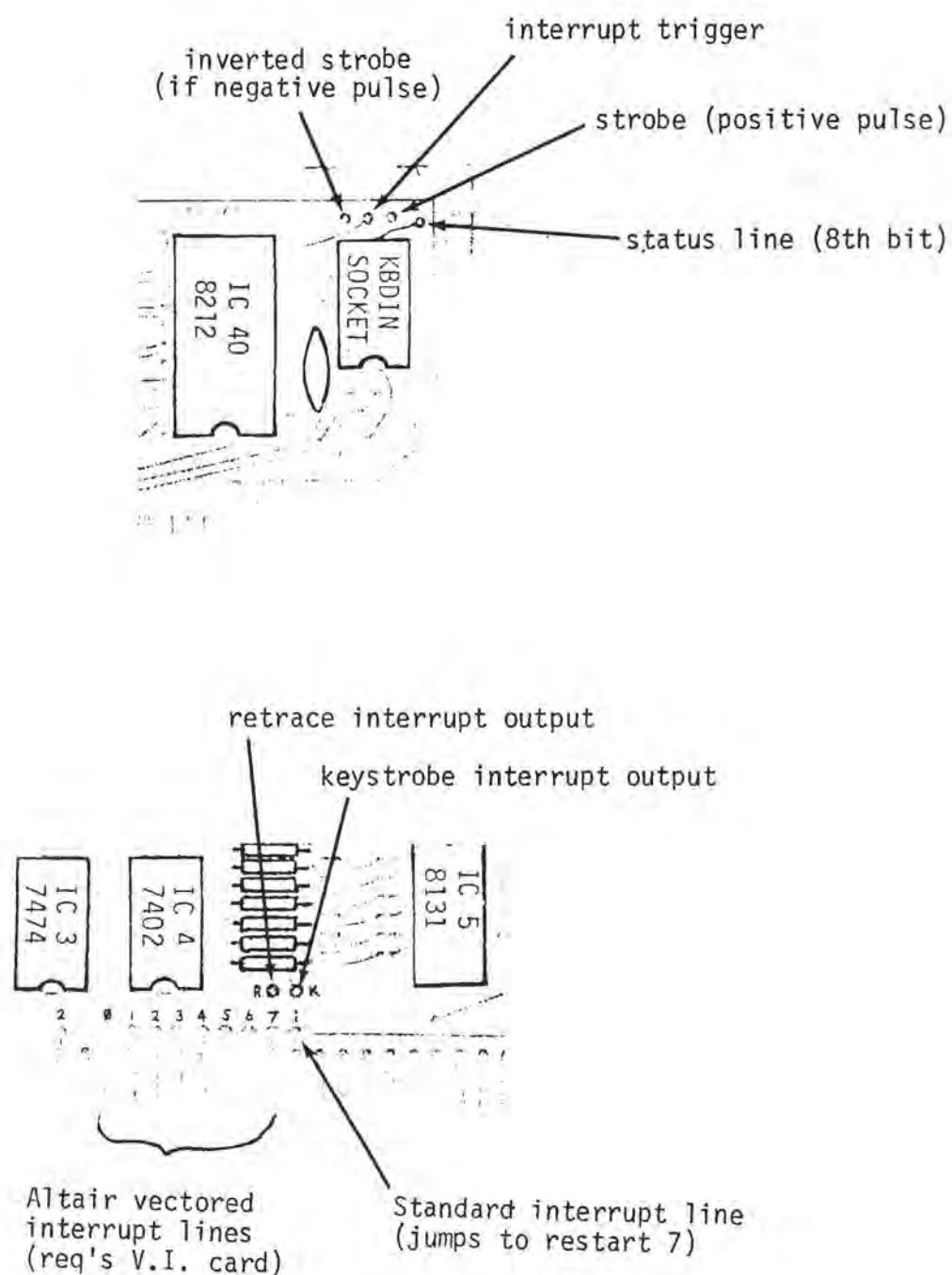


Figure 1-8

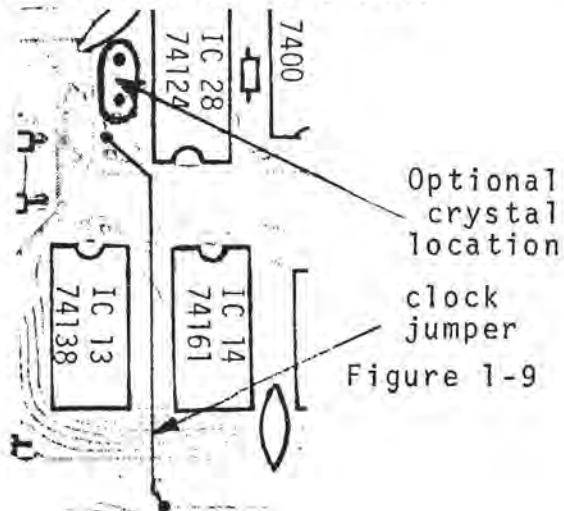
1.7.3 optional voltage regulator

Provision has been made for the optional negative voltage regulator required by a number of keyboards. The pads and traces for this voltage supply are located adjacent to the keyboard input socket, just above the large etched work "VIDEO." The supply regulates the -16V line by means of a resistor and zener diode stabilized by two capacitors. The four components are laid out in parallel, with the resistor nearest the socket, then the disc capacitor, the zener, and the tantalum capacitor in order below that. The choice of resistor and zener values depends on the voltage and current requirements of the keyboard.

1.8 Install optional on-board clock

Do not perform this step unless you are interfacing with a system that does not provide a clock.

The timing circuitry of the video terminal interface is synchronized to the 2 MHZ Altair system clock, which then determines the horizontal and vertical retrace rates. If the VTI is connected to some system other than an Altair-compatible bus, a 2MHZ clock may not be available. For this reason, provision has been made for an on-board clock.



Shown in Figure 1-9 is a region of the board adjacent to IC 28. By inserting a 2MHZ crystal (i.e., as close to 1.9445 as possible for maximum stability of image) in the location shown,

a 2MHZ TTL compatible clock will appear on pin 10 of IC 28. This can be jumpered to the bus clock input as shown. Since the on-board clock will then be driving the bus connector pin (pin 49), a board configured in this way should never be plugged into an Altair system, since opposing clocks will fight each other (to the death?). If the on-board clock is used and one wishes to be able to plug in to the Altair bus, simply cut the trace from edge connector pin #49 to the jumper pad adjacent to IC 3.

2. Theory of Operation

2.1 Block diagram

The principal functional blocks which form the video terminal interface are shown in figure 2-1. The on-board memory is connected in parallel with the keyboard input port to an array of I/O buffers driving the Altair data bus. This allows the transfer of information between the memory and the data bus or between the keyboard and the data bus. These data transfers are controlled by logic driven from the address and control lines. For example, the processor can read or write a location in memory just as it would with any main memory--it outputs the memory address (16 bits) while signaling a read or a write by the state of the control bus. The six most significant address bits are compared to the jumper selected bits (as discussed in section 22). If these bits match, then the remaining 10 address bits are gated through to select the memory location. At this time the

appropriate bus drivers are enabled to read from or write into memory, according to the control bus command. If the control bus signals neither a memory read nor a memory write, but rather an input instruction, then the keyboard buffer is enabled instead of the memory. Note that the input port address (8 bits) is the same as the most significant byte of the 16 bit memory address. When the processor is not accessing the video terminal interface with an input or memory instruction, then the video refresh circuitry takes control of the memory. The memory locations are scanned by the control and sync generator, with the memory data being fed into a character ROM. This read-only memory stores the video dot pattern of each ASCII character. The character font is a 7 X 9 matrix, so that each ASCII character has 9 memory blocks 7 bits wide in the ROM. Thus each line of characters on the TV screen results from many sequential scans through a line of memory locations. Each scan increments a counter so that the ROM reads off the next line of the dot matrix. Each clock of 7 bits read from the character ROM is loaded in parallel into a shift register and shifted out serially. This signal is then mixed with the video sync signals to form the composite video output.

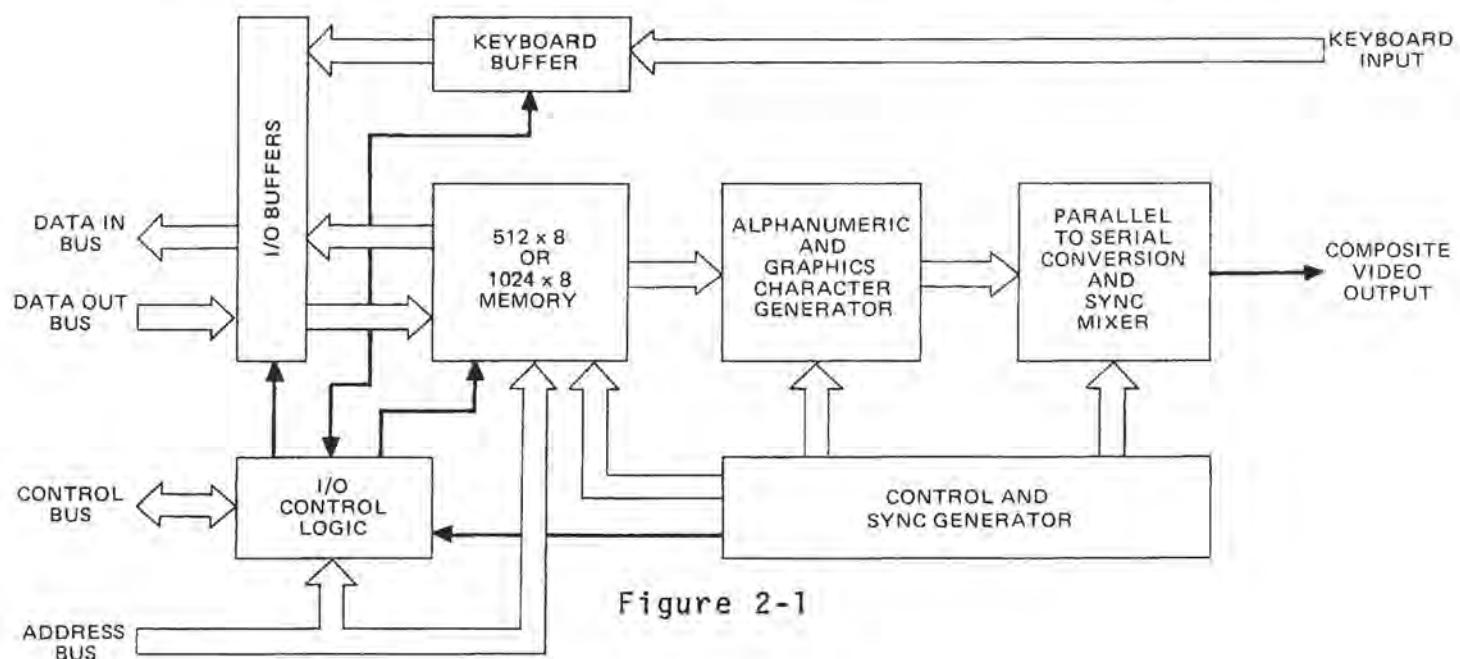


Figure 2-1

2.2 Schematic diagram

A more detailed view of the board circuitry is shown in the schematic diagram, Figure 2-2 (at end of volume). We are now going to examine the board in some detail to see how it performs its various functions. The level of complexity is fairly high; not all readers will find it useful.

Look at the schematic and note that all the on-board memory, data latches, and bus drivers are connected to a common on-board data bus. This bus can be driven by, or can drive, the Altair's own data bus. We will be referring to the video terminal interface (VTI) data bus as the on-board bus, and the Altair bus as the external bus.

Another point of terminology is sweep vs line. Each character on the TV screen consists of a selection of dots in a dot matrix that is seven dots wide by nine high, embedded in a field of ten

by fifteen dots (to provide space between characters). So the TV picture tube must sweep fifteen times to produce one line of characters.

The following discussion applies equally to the 32-character line and the 64-character line options.

2.2.1 symbol generation

With a low on the OE (output enable) line from IC10 to the RAM (random access memory) pins 9, the addressed portion of the RAM is continuously sent to the internal data bus in the refresh mode. Eight-bit display data on the internal data bus is sampled and held in the latch IC39 whenever there is coincidence (in IC29) of a dot pulse from the dot clock IC28 and an "end of character" (EOC) signal (tenth dot carry) from the "dot counter" IC14. In the absence of a one in the MSB (most significant bit) from the latch, MUX's (multiplexors) IC32 and IC35 pass the seven-dot conversion pattern of this display data from the character-generating ROM (read-only memory) IC36 to the 7 LSB (least significant bits) of the output shift register IC34. When the eighth bit specifies that graphics are being generated, these MUX's switch to select all ten bits of the data for the shift register from IC37 and IC38. IC37 and IC38 are, in effect, the graphics generation ROM.

In the case of non-graphics characters, the first three dots of every character space are always low to create spaces between letters. Note that, while the latched data for the nth

character position of the sweep is identical for fifteen consecutive sweeps, the ROM output may vary in each sweep, according to the additional addressing from the sweep counter half of IC16a. The sweep counter is self-resetting after every fifteenth sweep, and this resetting action is accumulated in the line counter half of IC16b.

In similar fashion, the dot counter IC14 is self-resetting every tenth dot, and its output is accumulated in the symbol counter IC17. The combination of line and symbol counter outputs determine the address of each individual character stored in the memory (IC's 20 through 27). Since all of these counters (dot and character, sweep, and line) are reset by appropriate relationships to the horizontal and vertical sync (respectively) of the TV raster, the lowest memory address will always contain the record for the top left corner of the TV display. Corresponding relationships are similarly maintained between other addresses in memory and positions in the display field.

2.2.2 raster & timing

Horizontal sync, vertical sync, and vertical blanking are timed by subcounting the absolute frequency system clock. Horizontal blanking is initiated at the end of sweep by subcounting the variable frequency dot clock IC28, and blanking is maintained by a variable-duration one-shot IC33a. Varying the "pos" pot changes the one-shot delay and thus the position in the next sweep where the display is again unblanked. Varying the dot clock

frequency ("width" pot) changes the rapidity with which the full line character count will accumulate to initiate horizontal blanking and therefore the distance across the screen that is used for display.

The system clock is divided by nine in IC2 and again by thirteen in IC1. This division by 117 of a 2 mhz clock produces a carry from the presettable binary counter IC1 every 58½ microseconds (17.094 khz). This carry on exit from the highest (16th) state (all four output bits = 1, or binary 15) is used to preload a binary 3 into the same IC1 so that it may again divide by 13. This binary 3 at the IC1 outputs will therefore last for one-thirteenth of the period between carries and is passed through IC4a to the TV for horizontal sync. The same carry triggers the horizontal blanking one-shot. The carry is also used to clock the 4-bit binary sweep counter (IC16a) which is used both to address the character generation ROM and to signal the line counter IC16b every fifteen sweeps that a new display line is being addressed.

When 16 line counts ($16 \times 15 = 240$ sweeps) have accumulated in IC16b, the carry resulting from the transition from its binary 15 state to its binary zero state is inverted by IC12 to set the vertical blanking flip-flop IC3. In addition to blanking the screen, IC3 also enables the 1 of 8 decoder IC13. After eight blanked sweeps have been counted by the sweep counter IC16, Pin 14 of IC13 will go low, producing a vertical sync pulse.

This vertical sync lasts the seven more lines until IC16a resets itself and advances the line counter. IC4 ANDs this vertical sync with the horizontal sync carry, so that the interruptions in the wide vertical sync pulse maintain horizontal sync.

Further subcounts of the sweep and advances of the line counter accumulate in IC16 a & b until IC13 decodes the 37th blanked sweep to trigger the pulse stretcher IC33b. (Line counter = 2 and sweep counter = 7.) IC33b is a very short duration one-shot which terminates the vertical blanking (disabling IC13) and also resets the sweep and line counters for top of the page addressing. The subsequent termination of horizontal blanking has the character counter IC17 reset to prepare all addressing from the top left of page as described below.

2.2.3 symbol and raster synchronization

Termination of the horizontal blanking one-shot IC33a reenables the dot clock oscillator IC28a but does not unblank the screen. At this time, symbol count addresses are set to zero, but the data latch IC39 contains unrelated data sampled with some previous address. Similarly, the shift register IC34 contains old data. The screen has been darkened by the dot blank flip-flops of IC31 which have been held set by the horizontal blanking. The symbol counter IC17 MSB is presenting a zero to the D input of flip-flop IC31a, however. After the first ten dots from the dot clock, the shift register (which is shift-clocked by dots) is emptied and the EOC (end-of-character)

signal from the dot counter IC14 sends load signals gated through IC29 to both the data latch and the shift register. Since propagation time through the ROM's and MUX's is not zero, the latch now contains beginning-of-line data, but the register is loaded with different but still useless data. The same end-of-character pulses, however, have advanced the symbol address in IC17 by 1 and have also propagated the zero at the input of the first D Blk (dot blank) flip-flop to the second flip-flop. The ROM and MUX paths present valid first symbol data to the shift register so that the second EOC pulse loads first symbol dots into the shift register and second symbol data into the latch. They also propagate the zero through the second dot blank flip-flop so that the screen is unblanked for the first symbol data shifted out of the register by the subsequent ten dots.

When the 32nd (or 64th) end-of-character pulse accumulates in the character counter, it loads the data latch with the 32nd (or 64th) character and the register with the next-to-last character. Simultaneously, the MSB of the symbol counter presents a 1 to the dot blank flip-flops, and the next 20 dots shift the last two symbols out to the video, and the 1 through the flip-flops to blank the screen in the 33rd (or 65th) character position. The dot clock runs, and the dot and symbol counters keep accumulating, but the MSB of the character counter maintains its 1 input to the dot blank flip-flops until either double the number of symbols is

counted or, as normally, horizontal sync and horizontal blanking occur to stop the dot clock, reset the symbol counter, and reaffirm the dot blank.

Clocked by the sweep counter reset, the line counter will increment every fifteen sweeps until the vertical blanking process described above resets the MSB's of the addressing system.

2.2.4 external bus and keyboard interfacing

The comparator IC5 compares the 6 MSB's of the external address bus with the jumper pattern selected for display memory addressing. If these agree at the time IC12d finds coincidence between system, sync (pin 76) and ϕ_1 of the system clock (pin 25), the board strobe (BS-) is output to enable the data bus line drivers and to switch the address MUX's (IC's 10, 18, and 19). In the switched condition, RAM address is determined by the ten LSB's on the external address bus instead of by the combination of line and symbol counters used in the display refresh mode. The BS-strobe also enables the line drivers that put internal data bus information onto the external data bus. If INP+ (pin 46) is also true, keyboard data latched in IC40 will be sent to the CPU via the line drivers. The MEMR+ signal, if present, similarly enables the memory output to the on-board bus. If MWR+ (pin 68) is high with BS-, the line receivers are enabled by IC9c to transfer the external data bus to the internal data bus and write it into the onboard RAM. In this

way, CPU data can be written into display addresses, keyboard data can be input to the CPU, or memory can be read from the display RAM addresses into the CPU. Keyboard data can be latched into IC40 in response to "key pressed" strobes of jumper selected polarity. Jumper provisions are made to copy this strobe on bit 8 of the keyboard input. A jumper pattern to pin 4 of the external bus permits sending an interrupt request to the CPU when the latch IC40 is updated by a "key pressed" strobe.

3.0 Troubleshooting

This discussion is intended to help the hobbyist who has some electronics troubleshooting experience. Those with no experience should not attempt troubleshooting. If a synchronized array of characters cannot be achieved by adjustments of sync controls on the CRT (or TV), check first for the more obvious and frequently encountered problems. Most typical will be such items as:

1. Loose connections to system or to display.
2. Improper interfacing to display's video input (biasing, etc.).
3. Omission or improper installation of components on the board (reversed diode or chip orientation).
4. Soldering problems of unsoldered contact or solder-bridge shorts.
5. Omitted or wrongly selected jumper patterns (line length, address selection, etc.).

The discussion below follows one of many possible logically sequenced procedures to localize problems and is written for those without access to an oscilloscope.

Start with a good visual inspection of connections and of the board itself. Progress through checks on the power supply busses and video output to electrical test patterns of the signals on the board. In using the elctrical test patterns, work from end results backwards towards those parts of the circuit which contribute to the end results. For example: if the proper raster sync signals are doing their job, all further measurements concerning these circuits involved can be omitted in favor of checking contributions to character presentation.

3.1 Power mains

3.1.1. If visual inspection looks good, see if the power mains are proper. There should be $+5.0 \pm 0.2$ VDC on the VCC bus.

Convenient clip lead points include:

A. Ground reference: the metallized board area under the voltage regulator heat sink at the top left is a good one. The board has been designed with a blank area on the reverse side so that the other jaw of a clip cannot short any signals there.
(Watch out for this at other locations!)

B. 5 volt bus: the right leg of the LM340T regulator. A voltage below tolerance here may indicate either a heavy current load from a misconnection or a reverse-oriented IC or that your power main feeding the board has less than 7 volts available.

Zero volts at this point probably indicates missing power to the board (a cold regulator) or a dead short on the board in which case the regulator will be very hot to touch. (Don't panic. You will be amazed at its recuperative capability when the short is cleared.)

C. VDD bus for the character generating ROM IC36: Measure $+12V \pm 10\%$ at the junction of R20/C29.

D. VBB bus for IC36: Measure $-3V \pm 10\%$ at junction Z1/C30. (This is the only negative voltage.)

3.1.2 If power bus shorts are suspected, ohmmeter verification involves considerations of the polarity of the test leads. The board will not suffer from checks where the ohmmeter leads apply the polarity expected from the power supply and an open circuit voltage not exceeding the power supply value. The non-linearity of the load prevents us from predicting what an unknown ohmmeter will read on a normal board, but readings below an ohm mean that you should look for a short or an inverted IC. Reverse polarity from ohmmeter leads can be damaging unless the current is limited to low values. Most series-connected 50 micro-amp movement VOM's are safe when only the 1.5 volt battery is used on the scale selected.

3.2 Signal tracing

Unsolder the right end of the 200 ohm R18 (junction with pins 2, 4, 6, & 8 of IC30) and attach a clip lead to the free end of the resistor for use as a scope probe. (Keeping a wire in the hole

what is missing. If, in the "normal" connection (i.e: lead clipped to where R18 should be soldered), there is an array of bright and dark spots on the display, chances are that video is being generated and that you will be chasing sync or blanking troubles. With only video coming through, most CRTs will at least partially sync on the video itself, and patient tinkering with the sync controls on the display and the two pots on the video board should give at least some torn-up version of what is trying to be a display. If you have sophisticated your power-up sequence to program a blank display, either alter the sequence until troubles are cured or remove programming to the board. Random states in the board RAM at power-up will produce some interpretable static pattern. But maintain the system clock connection. Horizontal sync is derived from that clock. (The board is testable with nothing more than proper power supplies and a 2 mhz clock for inputs.)

IC34-6

No video pattern? Let us see if it is shifting out of the register IC34-6 (pin 6 of IC34). Got it? Then the path through IC30 is not passing it. Check for it at the input pin 9 and output pin 8 of IC30. Following the path should reveal a gap in signal passage that is correctable. This is the concept of signal tracing that will be assumed throughout the remaining discussion.

No video shifting out of IC34-6? Well, is there data on the input pins to be loaded for shifting - or a load signal to load it - or a dot clocking to shift it out?

for the right end of R18 makes an easy way to remake the "normal" connection with the clip lead.)

DC voltages would normally read 1.6 V at this junction, but, when open, the clip lead will read about 4.5 and the IC30 pins less than 0.1 V. This produces DC levels at the 2N5449 emitter of about 2V normally (average of normal waveform) and near 4V with an open test lead. 27% of these values should be found on the cable to the CRT. (If you have D.C. coupled into your CRT video, check that your design is proper for these values.)

Those users owning oscilloscopes probably have sufficient technical background to interpret the following discussion into equivalent scope presentations. This discussion assumes that the only signal tracing display available is the TV or CRT intended for computer display use. Therefore, the first checks are that the output stage is functioning and that its responses are visible on the CRT. If NOGO on these, check your cable and CRT input arrangement.

3.2.1 video interface

Grounding the probe lead should pull the output emitter down to around a volt, and opening it should give a rise to around 4v. This transition should couple through the AC coupling to your CRT and be apparent as momentary brightening as the lead opens.

3.2.2 localizing on the video path

If logic levels applied to the clip lead are modulating the display brightness, but you are having to troubleshoot, let us consider

First the dot clock on IC34-9: This should show as a raster full of tiny white dots. Depending on the setting of the "width" pot, there should be from 100 or so to almost 900 on each raster sweep, but several factors influence this. Sync and blanking, if they are working, keep many dots out of the visible area. Also, the bandwidth of this setup may not permit you to discern dots at the higher frequency settings of the dot clock. Best to view this at the minimum frequency (ccw) setting of the "width" pot (pot at top left). Do not bother counting dots. Their presence is all that is necessary to show register shift clocking input. Since this signal is negative true, a brighter presentation may be found at the inverted form IC29-8 on IC29-8. Absence of sync should not prevent this display from being recognizable.

EOC (end-of-character) loading signals on IC34-7 should show as dark (negative true) vertical bars every tenth dot (except for a portion of the screen where horizontal blanking normally disables the dot clock). Their presence proves the dot clock (and dot counter) whether we check IC34-9 or not. The number of bars visible is variable by the dot clock frequency ("width" pot) and by the "pos" pot control of sweep blanking. Although the blanking path is broken by lifting R18, the composite sync path is not. Therefore, if a strong sync is at work, some of the display, such as the area unbroken by vertical bars, may be sync'd into times not visible on the screen. This point about

sync must be borne in mind as you check many of the waveforms - particularly in the sync path itself.

Assuming that shift (dot) clocking and its subcount, EOC load clocking, are available, is there video data on the input pins to be loaded? Each of pins 1 through 5 and 11 through 14 should show a screen pattern of white and dark states as wide as the distance between the vertical bars seen on pins carrying the EOC or shift loading pulses. So too should input and output pins of the MUX's IC32 and IC35. Also the outputs the ROM IC36 and the graphics generators IC37 and IC38.

IC34-1
to 5 &
11 to 14

The patterns associated with outputs from IC37 and IC38 have a right to change every 5 sweeps. At the IC39 inputs to the display generators IC36, 37, & 38, however, the sweep patterns should not change more frequently than every fifteenth sweep. These last patterns show what the memory is requesting for each character position of ten dots by fifteen sweeps. Counting these dimensions is generally not necessary. Merely noting that the fineness of detail is less at the input to generators than at the output is usually sufficient for trouble localizing. Subcounting is discussed under 3.2.3 and 3.2.5.

The screen pattern for any significant bit input to the generators should be traceable back through corresponding pins of the sampling latch IC39 to the same significant bit of the internal data bus. But remember, the nth character in memory is held

IC39
data bus

on the bus, while the n-1th character is held in the latch until an EOC pulse strobes the latch and increments the memory address. If sync and clocking are at work to keep the display pattern straightened up, any lack of correspondence of the patterns up the path can be discerned. Without sync, it may take both a photographic memory and a lot of luck -- but the chances are that you would not be needing that level of detailed troubleshooting without sync, anyway.

In like fashion, grounding pin one of IC32 forces MUX's IC32 and IC35 to select only graphic symbols from IC37 and IC38. This change is most apparent with a sync'd display, but some shift should usually be discernible in the pattern for any shift register input pin. The degree of change will depend on how frequently the MSB is a one in the RAM. Correspondingly, the display probe on IC32-1 will show which memory locations contain graphics or non-graphics characters. An MSB in memory is inverted in the latch to select graphics.

3.2.3 localizing on the EOC (end of character) path

If you had dot clock input to shift register IC34-9 but no stobe (IC34-7) to load the register, you will want to check back to where the EOC is generated by counting every tenth dot in IC14. In fact, failure of IC29 or other problems can permit it to count by other than ten, with some weird results in displays.

Clock dots are discernible at the input IC14-2. Slowing the dot clock (CCW on the "width" pot) makes these countable by eye. A

piece of paper on the screen or a millimeter scale may help. Sync helps here but should not be necessary to array the pattern of dots into vertical bars. IC14-14 has half as many vertical bars but of double width. Pin 13 has narrow vertical white bars equal to twice the width of the bars on pin 14. The total pattern of pin 13 is repetitions of black, white, black, white, white vertical bars. The last two whites show as a double width white as the carry preloads a 6 into this 4 bit binary counter. This preload makes it produce a carry every tenth dot. If pin 13 looks right, chances are that all the rest is okay.

The tenth dot carry on IC14-15 is the EOC (end of character) signal. It should appear at the input to the symbol counter IC17-13. An inverse (negative true) of this pattern should be found as loading signals n latch IC39-11 and shift register IC34-9. Of course, if there is no dot clock, none of this paragraph is working properly. On the other hand, presence of dots anywhere does not leave much room for problems in the dot clock.

3.2.4 Localizing on the dot clock path

If either the shift register or the dot counter is getting dots, you are in for some detail checks of solder bridges to ground, a single NAND gate in IC29, or some such, because the clock is present at the other end of these places. If neither is present (and of course no EOC signals), then look for dots at the clock IC28-7 IC28-7. Using a voltmeter, check its "width" pot for the ability

to vary IC28-2 from zero to 5 volts. Check also for the enabling portion of the horizontal blanking signal on IC28-6. This may be hard to see as a broad vertical bar in the presence of strong horizontal sync, but if desyncing gives you a torn version of it, it is probably okay. A voltmeter reading on IC28-6 of 5 VDC would be a continuous disable signal. Under proper conditions, the average of the horizontal blanking waveform reads typically 0.9 to 2.3 VDC on a meter at IC28-6. The value is under control of the "pos" pot which varies the time delay (and thus the average DC value) of the blanking monostable.

3.2.5 Localizing on the horizontal blanking path

Under the most ideal conditions of sync and blanking, events occurring during flyback, retrace, or blanking should not be visible. Note that opening R18 does not open the composite sync path at IC30-10. Therefore, sync, if operating, will reach the CRT sync circuits - regardless of what is done with the probe lead. Remember, even without sync working, most CRT's or TV's will find in many of the test signals something repetitious enough to sync on. There is usually a way to view sync-hidden signals by misadjusting the horizontal hold control of the CRT to force a "tear" in the picture. Then if the sweep rate is calibrated in time units, the signal can be measured in the torn portion. An example of this is horizontal blanking. Forcing a torn but stable pattern reveals a dark space in each sweep when looking at IC28-6. Varying the "pos" pot changes the width of the space.

Typical values from stop to stop on the pot are about 10 or 20 microseconds (see section 3.3.1 on time calibration) but, if you can change it, it is working. Perhaps easier to see is its inverse - a logic high on IC33-5. For this, you should not have to force the tear. Horizontal blanking that is high logic will appear as a bright vertical bar at one or both sides depending on where the CRT is syncing. For most IC's, if Q is working, \overline{Q} probably is also. Take the easiest way down the localizing path first and back up to the harder ones only when necessary.

No horizontal blanking? How about the horizontal sync which triggers the IC33 monostable multi-vibrator to stretch the sync into a wider blanking? The carry out of counter IC1-15 should have its inverse on IC33-9. This is a $4\frac{1}{2}$ microsecond pulse every $58\frac{1}{2}$ microseconds if your system clock is 2.0 mhz. Actual horizontal sync is the same width, but $4\frac{1}{2}$ microseconds later, and can be seen on IC4-13. Its inverse is on IC4-1 but is also mixed with vertical sync. Observation of a once-per-sweep, narrow vertical bar is probably sufficient to eliminate further details up this path, but if things are not clearing up, you may want to calibrate time as in 3.3.1.

If these are NOGO, is the system clock on edge pin 49 and is it reaching IC2-2? The leading edges of the white bars are $\frac{1}{2}$ microsecond apart for a 2.0 mhz clock at IC2-2 and $4\frac{1}{2}$ microseconds apart at the carry on IC2-15. Or is it dividing by nine? clock

You can use your piece of paper or plastic millimeter scale to ratio the distance between leading edges of the bars. However, if the vertical bar pattern on IC2-14 is repetitions of black, white, black, white, black, white, black, white, white, then the binary 7 is apparently preloading on every carry and division is probably okay. (Compare this with the discussion of the dot counter in 3.2.3.)

IC2-14

Counting bars will only tell you how many of the 58½ microseconds per sweep are visible on your CRT and usually does not contribute to trouble analysis.

IC1-2 has an inverted form of IC2-15 showing a dark bar every 4½ microseconds, but division by 13 is difficult to ratio unless you have a rare CRT that has a horizontal width control that permits shrinking the picture sufficiently to see both ends of the sweep. But then - if any of IC1-11, IC1-12, IC1-15, or IC4-13 have an observable once-per-sweep bar, horizontal sync seems to be doing its job.

IC1-2

IC1-1

3.2.6 sweep and symbol related counter patterns:

Verification of sweep counter test patterns is difficult in the absence of horizontal sync. Since the sweep counter is counting the carries from the same counter that generates horizontal sync, the presence of one signal without the other would indicate that the integrity of any missing path should be reestablished before proceeding. The clocking input IC16-1 is a once-per-sweep pulse which may not be in the visible portion

IC16-

of the sweep unless a tear is forced in the horizontal hold. All other patterns are stretched by the sweep into horizontal bar patterns with the exception of the reset IC16-2. The reset is like the clock on IC16-1, except a) it occurs every 15th sweep; b) it is a $4\frac{1}{2}$ microsecond darkening instead of a brightening; and c) it occurs $4\frac{1}{2}$ microseconds later (to the right) on the screen. It is therefore probably visible only under torn conditions.

Correct patterns for pins 3, 4, 5, and 6 of IC16 can be inferred from the timing diagrams. A quick check of proper operations and counting by fifteen can be made on pin 4. The pattern for IC16-4 is: every other pair of sweeps is white (2nd, 4th, and 6th pairs) followed by the single white 15th sweep during which the counter is reset. Symbol lines are perhaps better defined by the double black sweeps visible on IC16-13. These occur because of the adjacency of the first and last sweeps, which are both dark, while all even numbered sweeps including those during retrace are bright.

As further subcounting is done in the line counter, IC16-11 shows every other line (group of 15 sweeps) as dark or bright. Forcing a tear in the horizontal sync can permit staggering the gap caused in each sweep. This can permit an alternate form of checking division by 15 (sweeps per line) in the sweep counter. The MSB in the line count is white in the bottom half of the display. After the bottom bright trace of IC16-8, IC3-9

IC16-8

shows the bright inverse of 8 sweeps of vertical blanking at IC3-9 the bottom of the screen and the later sweeps normally hidden by the vertical blanking at the top of the screen.

Patterns for the symbol counter IC17 can be directly inferred from the theory discussion and the pin outs of the 74393. The EOC pulses described in 3.2.3 are seen as a vertical bar per symbol IC17-1 space on IC17-13. Successive divisions by 2 on pins 11, 10, 9, 8, 3, 4, and (if 64 symbol option, pin 5) are seen as fewer, wider bars. Reset will appear on pins 12 and 2 as it does at IC33-5. (Refer to Section 3.2.5.)

The functions of IC13 and IC33b are not directly observable in the presence of sync. If no sync at all is reaching the raster, normal operation of IC33-13 can be noted as small (on the order of 30 nanosecond) specks scattered in regular fashion throughout the raster. If sync is working, operation may be inferred by noting rapid regular jumping of vertical sync when IC33-1 is held to ground.

The combination of IC33b and IC13 can be checked by grounding IC13 pins 4 and 5 of IC13. Under this condition, the normal output connection to the display will show repetitions of seven darkened sweeps of vertical blank followed by thirty visible sweeps of retrace allowance. Also, placement of the test clip on IC13-12 will show continuous repetitions of seven dark sweeps, eight white sweeps, seven dark, fifteen white.

The outputs of the symbol and line counters should show obvious $\div 2$ relationships for ascending orders of bits. These patterns should be traceable through the MUX's IC's 18, 19, and 10 and decoder IC11 to the associated RAM address input pins.

IC18,19
10,11

Normal events on the dot blank flip-flops IC31-2, 4, 5, and 8 produce vertical bars on a once per sweep basis. Position and width of the bars is variable by both "pos" and "width" pots. The waveform average of these waveforms read on a DC meter will also vary under control of these pots. If sync prevents visual observation of these pulses, DC voltage variations by the pots can be taken as proof that the variable width dot blank is reaching the right places.

3.3 Diagnostic aids

Viewing the display in normal conditions gives information on where to start troubleshooting. A blank screen directs attention to sections 3.2.1 through 3.2.5, which look for dynamically changing patterns originating in a sequentially scanned memory, being translated in the ROM's and being shifted out of the register. In the process, dot clocking and EOC signals are investigated as necessary.

A dynamic but useless display in normal conditions, on the other hand, directs attention to the subcounters and decoders which control memory address, the blanking of the display borders, and the orderliness of symbol element display.

Thoughtfully examining the display can give valuable clues for trouble localizing. Torn-up symbols logically relate to the sweep counter and its derivatives in the line counter and vertical blanking. Wrong symbol displays indicate a need to also verify dynamic signal paths between symbol and line counters, or the ability to load memory properly. Since many of these are interrelated in unpredictable syndromes, it is impractical to anticipate all combinations here. Problems relating to data exchanges between the memory and/or keyboard and the system CPU are not peculiar to the video display and should be approached in whatever is your standard method for handling problems with memory or peripherals.

3.3.1 time calibration

In verifying the timing diagrams related to horizontal sweep rates, the $4\frac{1}{2}$ microsecond wide bars on IC2-14 give a quick idea of how much of the timing diagram will show on your TV. A 50 microsecond block is indicated on most of the timing diagrams, but a typical TV might show five white and give black bars on IC2-14 for a total display of 45 microseconds. Remember also that horizontal sync may permissibly vary widely, so that your picture may start at a different point in comparison to the arbitrary marks on the diagrams.

IC2-14

Calibration of the vertical dimension or vertical sweep time base is perhaps easiest by looking at IC16-3. The leading edges (measuring top to bottom) of the groups of white sweeps are

15 sweeps or 877 microseconds apart. A 16 line (240 sweep) visible raster is 14.04 milliseconds, and vertical sync recurs every 277 sweeps or 16.205 milliseconds.

Occasionally, an integrated circuit is itself defective. You can sometimes determine this by swapping ICs from one location on the board to another -- i.e., ICs that are used in more than one location (like memory). If you find that you were supplied with a defective chip, it will be replaced free (see the warranty information sheet included herein).

4. Software

4.1 Video terminal software

Both the input to and the output from a computer is ordinarily a string of characters, whether it be characters typed in from a typewriter-like keyboard or output from the computer to a printer. Not all of these "characters," however, strictly correspond to a printed symbol, like a letter. Consider the output to a printer. Some "characters" will cause the printer to perform some function other than a keystrike -- such as carriage return or backspace.

The VTI is essentially a block of memory, and at the hardware level does not distinguish between characters and other functions. Without an intervening program, the VTI would send a "carriage return" on to the screen as a character, rather than returning to the beginning of the line.

We include here a program that accepts a string of ASCII characters and causes them to appear on the screen exactly as

the characters would be printed by a printer. "Carriage return" causes the cursor to return to the beginner of the line, "line feed" causes it to move down one line, and so forth.

The program includes a keyboard input routine, which puts the characters you type on the keyboard directly onto the screen, with proper carriage return, line feed, and other functions. Load the program as written. To use the computer as a "TV typewriter," connect the keyboard to the parallel input port provided on the video board, using the Altair interrupt.

(See Section 1.7.2.)

This program when executed at address ZERO causes characters typed in at the keyboard to appear on the screen as they would be printed by a printer.

The principal usefulness of the program is to interpret the output of another program which would ordinarily be sent on to a printer, so as to put the appropriate visual display on the screen.

Programs ordinarily send a character from the accumulator to a serial output port in response to the instruction "out". The following program includes a subroutine called "out," located at hexidecimal address 1D00. When called, this subroutine interprets the character in the accumulator as required to put it on the screen. In converting a program to run with the VTI, substitute "call out" for the output instruction.

Hexidecimal Address	Op Code	Mnemonic Instruction	Comments
0000		0100 SCRN EQU 8800H	*VIDEO SCREEN ADDRESS
0000		0110 STR EQU 1CFFFH	*STORAGE FOR SYMBOL UNDER CURSOR
0000		0120 STS EQU 1CFEH	*STORE OUTPUT MODE
0000		0130 CURS EQU 1CFCH	*STORE RELATIVE CURSOR LOCATION
0000		0140 SEND EQU 8CH	*1ST BYTE OF SCREEN END
0000		0150 LINE EQU 64	*LINE LENGTH
0000		0160 CS EQU 0FFH	*CURSOR SYMBOL (CRUB OUT)
0000		0170 LT EQU 3FH	*LINE TERMINATION CHARACTER
0000		0180 KBD EQU 88H	*KEYBOARD PORT ON VTI
0000		0190 ORG 0000	
0000 21 00 00		0200 LXI H, 0	
0003 22 FC 1C		0210 SHLD CURS	
0006 7D		0220 MOV A, L	
0007 32 FE 1C		0230 STA STS	*SET UP WITH CLEAR SCREEN
000A 21 11 00		0240 LXI H, LOOP	*AND CURSOR AT UPPER RIGHT
000D E5		0250 PUSH H	*USER MUST DEFINE OWN STACK AREA
000E C3 65 1D		0260 JMP FF	
0011 FB		0310 LOOP EI	
0012 C3 11 00		0320 JMP LOOP	
0015		0330 ORG 38H	*RESTART ?
0038 DB 88		0340 IN IN KBD	*INTERRUPT DRIVEN KEYBOARD
003A F6 88		0345 ORI 88H	
003C F6 88		0350 ORI 88H	
003E 47		0360 MOV B, A	
003F CD 00 1D		0370 CALL OUT	
0042 78 9		0380 MOV A, B	
0043 C9		0400 RET	
0044		0500 ORG 1000H	
- 1000 2A FC 1C		1000 OUT LHLD CURS	
1003 EB		1010 XCHG	*PUT RELATIVE CURSOR IN D
1004 21 00 88		1020 LXI H, SCRN	*PUT SCREEN BLOCK ADDRESS IN H
1007 19		1030 DAD D	*GET ABS CURSOR LOCATION
1008 47		1040 MOV B, A	
1009 3A FF 1C		1050 LDA STR	
100C 77		1060 MOV M, A	*PUT BACK CHAR UNDER CURSOR
100D 78		1070 MOV A, B	*CHECK*
100E FE 88		1100 CPI 88H	*CTL H FOR HOME
1D10 CA 5C 1D		1110 JZ HOME	
1D13 FE 85		1120 CPI 85H	*CTL E FOR ERASE
1D15 CA 65 1D		1130 JZ FF	
1D18 FE 92		1140 CPI 92H	*CTL R FOR RIGHT
1D1A CA 74 1D		1150 JZ HT	
1D1D FE 95		1160 CPI 95H	*CTL U FOR UP
1D1F CA 7C 1D		1170 JZ VT	
1D22 FE 80		1180 CPI 80H	*CTL L FOR LEFT
1D24 CA 91 1D		1190 JZ BS	
1D27 FE 84		1192 CPI 84H	*CTL D FOR DOWN
1D29 CA E8 1D		1194 JZ LF	
1D2C FE 98		1200 CPI 98H	*CTL X (DELETE CHAR)
1D2E CA 99 1D		1210 JZ RO	

1D31 FE 89	1220 CPI 89H	*CTL I FOR INSERT (SET I/D)
1D33 CA 86 1D	1230 JZ SID	*CTL T FOR TEXT (X I/D)
1D36 FE 94	1240 CPI 94H	*CTL F FOR FEED (SET ALF)
1D38 CA B1 1D	1250 JZ RID	*CTL N FOR NORMAL TTY (X ALF)
1D3B FE 86	1260 CPI 86H	
1D3D CA BC 1D	1270 JZ SRLF	
1D40 FE 8E	1271 CPI 8EH	
1D42 CA C7 1D	1272 JZ RALF	
1D45 FE 93	1280 CPI 93H	
1D47 CA D2 1D	1290 JZ SSC	
1D4A FE 90	1300 CPI 90H	
1D4C CA DD 1D	1310 JZ RSC	
1D4F FE 8A	1320 CPI 8AH	
1D51 CA E8 1D	1330 JZ LF	
1D54 FE 8D	1340 CPI 8DH	*LINE FEED
1D56 CA 21 1E	1350 JZ CR	*CARRIAGE RETURN
1D59 C3 45 1E	1360 JMP DEF	*ANY OTHER CHARACTER
1D5C 21 00 00	2000 HOME LXI H,0	*HOME CURSOR
1D5F 22 FC 1C	2010 SHLD CURS	
1D62 C3 6F 1E	2020 JMP OUT1	
1D65 21 00 88	2030 FF LXI H,SCRN	*FORM FEED
1D68 36 3F	2050 WIPE MVI M,LT	*LINE TERMINATION CHAR 7FH
1D6A 23	2060 INX H	
1D6B 7C	2070 MOV A,H	
1D6C FE 8C	2080 CPI SEND	*SCREEN END?
1D6E C2 68 1D	2090 JNZ WIPE	
1D71 C3 5C 1D	2100 JMP HOME	*CLEAR, GO HOME
1D74 13	2110 HT INX D	*CURSOR RIGHT
1D75 EB	2120 XCHG	
1D76 22 FC 1C	2130 SHLD CURS	
1D79 C3 6F 1E	2140 JMP OUT1	
1D7C 21 C0 FF	2150 VT LXI H,0-LINE	*CURSOR UP
1D7F 19	2160 DAD D	
1D80 22 FC 1C	2170 SHLD CURS	
1D83 C3 6F 1E	2180 JMP OUT1	
1D86 3A FE 1C	2190 SID LDA STS	*SET I/D MODE
1D89 F6 01	2200 ORI 01H	*RIGHT BIT =1
1D8B 32 FE 1C	2210 STA STS	
1D8E C3 6F 1E	2220 JMP OUT1	
1D91 1B	2230 BS DCX D	*CURSOR LEFT
1D92 EB	2240 XCHG	
1D93 22 FC 1C	2250 SHLD CURS	
1D96 C3 6F 1E	2260 JMP OUT1	
1D99 3A FE 1C	2270 RD LDA STS	*RUB OUT IF I/D SET
1D9C 1F	2280 RAR	
1D9D D2 91 1D	2290 JNC BS	
1DA0 23	2300 SWAP INX H	*DEL CHAR, SWAP LINE IN
1DA1 7E	2310 MOV A,M	
1DA2 2B	2320 DCX H	
1DA3,77	2330 MOV M,A	
1DA4 23	2340 INX H	
1DA5 7D	2350 MOV A,L	
1DA6 E6 3F	2360 ANI 3FH	

1DAB C2 A0 1D	2370	JNZ SWAP	
1DAB 2B	2380	DCX H	
1DRC 36 7F	2390	MVI M, 7FH	
1DRE C3 6F 1E	2400	JMP OUT1	
1DB1 3A FE 1C	2410	RID LDA STS	*RESET I/D MODE
1DB4 E6 FE	2420	ANI 0FEH	*RIGHT BIT =0
1DB6 32 FE 1C	2430	STA STS	
1DB9 C3 6F 1E	2440	JMP OUT1	
1DBC 3A FE 1C	2450	SALF LDA STS	*SET ALF MODE
1DBF F6 40	2460	ORI 40H	*2ND BIT LEFT =1
1DC1 32 FE 1C	2470	STA STS	
1DC4 C3 6F 1E	2480	JMP OUT1	
1DC7 3A FE 1C	2482	RALF LDA STS	*RESET ALF MODE
1DCR E6 BF	2484	ANI 0BFH	*2ND BIT LEFT =0
1DCC 32 FE 1C	2486	STA STS	
1DCF C3 6F 1E	2488	JMP OUT1	
1DD2 3A FE 1C	2490	SSC LDA STS	*SET SCROLL MODE
1DD5 F6 80	2500	ORI 80H	*LEFT BIT =1
1DD7 32 FE 1C	2510	STA STS	
1DDA C3 6F 1E	2520	JMP OUT1	
1DDD 3A FE 1C	2530	RSC LDA STS	*RESET SCROLL MODE
1DE0 E6 7F	2540	ANI 7FH	*LEFT BIT =0
1DE2 32 FE 1C	2550	STA STS	
1DE5 C3 6F 1E	2560	JMP OUT1	
1DE8 21 40 00	2570	LF LXI H, 64	*LINE FEED
1DEB 19	2580	DAD D	*ADD 64 TO REL CURSOR
1DEC 3A FE 1C	2590	LDA STS	
1DEF 17	2600	RAL	
1DF0 DC F9 1D	2610	CC SCRL	*CHECK SCROLL
1DF3 22 FC 1C	2620	SHLD CURS	*UPDATE CURSOR LOCATION
1DF6 C3 6F 1E	2630	JMP OUT1	
1DF9 7C	2640	SCRL MOV A, H	*SCROLL ROUTINE
1DFA FE 04	2650	CPI 4	*OFF PAGE?
1DFC D8	2660	RC	*IF NOT, DO NOTHING
1DFD E5	2670	PUSH H	
1DFE 11 00 88	2680	LXI D, SCRN	*TAKE IT FROM THE TOP
1E01 21 40 88	2700	LXI H, SCRN+LINE	
1E04 7E	2710	SWP MOV A, M	*GRAB CHARACTER
1E05 23	2720	INX H	
1E06 EB	2730	XCHG	*GET ADDRESS ONE LINE UP
1E07 77	2740	MOV M, A	*PUT CHARACTER THERE
1E08 23	2760	INX H	
1E09 EB	2770	XCHG	
1E0A 7C	2780	MOV A, H	
1E0B FE 8C	2800	CPI SEND	*SCREEN FINISHED?
1E0D C2 04 1E	2810	JNZ SWP	*TAKE NEXT CHAR IF NOT
1E10 EB	2812	XCHG	
1E11 06 3F	2814	MVI B, LT	*BLANK LAST LINE
1E13 70	2816	LAST MOV M, B	
1E14 23	2820	INX H	
1E15 7D	2830	MOV A, L	
1E16 FE 00	2840	CPI 0	
1E18 C2 13 1E	2850	JNZ LAST	

1E1B E1	2860	POP H	*GET BACK REL CURSOR
1E1C 11 C0 FF	2862	LXI D, 0-LINE	
1E1F 19	2864	DAD D	*MOVE UP ONE LINE
1E20 C9	2870	RET	
1E21 3A FE 1C	2890	CR LDA STS	*CARRIAGE RETURN
1E24 1F	2900	RAR	
1E25 DA 32 1E	2910	JC BACK	*INSERT/DELETE? IF SO, DON'T
1E28 36 3F	2920	SLOP MVI M, LT	*SCRATCH END OF LINE
1E2A 23	2930	INX H	
1E2B 3E 3F	2940	MVI A, 3FH	*MAKE 1FH FOR 32 CHAR LINE
1E2D A5	2950	ANR L	
1E2E C2 28 1E	2960	JNZ SLOP	
1E31 2B	2970	DCX H	
1E32 3E C0	2980	BACK MVI A, 000H	*GO TO BEGINNING OF LINE
1E34 A3	2990	ANR E	
1E35 5F	3000	MOV E, A	
1E36 3A FE 1C	3020	LDA STS	
1E39 17	3030	RAL	
1E3A 17	3040	RAL	
1E3B DA E9 1D	3050	JC LF	*CHECK AUTO LINE FEED
1E3E EB	3052	XCHG	
1E3F 22 FC 1C	3055	SHLD CURS	
1E42 C3 6F 1E	3060	JMP OUT1	
1E45 3A FE 1C	4000	DEF LDA STS	*DEFAULT ROUTINE, CHECK I/D
1E48 1F	4010	RAR	
1E49 DC 5C 1E	4020	CC INSR	*INSERT IF NOTED
1E4C 70	4030	MOV M, B	*STUFF CHARACTER
1E4D 13	4040	INX D	*INCREMENT CURSOR
1E4E EB	4050	XCHG	
1E4F 3A FE 1C	4060	LDA STS	
1E52 17	4070	RAL	
1E53 DC F9 1D	4080	CC SCRL	*CHECK SCROLL
1E56 22 FC 1C	4090	SHLD CURS	*UPDATE CURSOR
1E59 C3 6F 1E	4100	JMP OUT1	
1E5C E5	4200	INSR PUSH H	*MAKE SPACE FOR INSERT
1E5D 7E	4210	MOV A, M	
1E5E 3A FF 1C	4220	LDA STR	
1E61 77	4230	MOV M, A	*REPLACE CHAR UNDER CURSOR
1E62 23	4240	SHFT INX H	*MOVE LINE OUT
1E63 4E	4250	MOV C, M	
1E64 77	4260	MOV M, A	
1E65 3E 3F	4270	MVI A, 3FH	
1E67 A5	4280	ANR L	
1E68 79	4290	MOV A, C	
1E69 C2 62 1E	4300	JNZ SHFT	
1E6C 77	4310	MOV M, A	
1E6D E1	4320	POP H	
1E6E D9	4330	RET	
1E6F 2A FC 1C	8000	OUT1 LHLD CURS	*KEEP CURSOR ON SCREEN
1E72 7C	8010	MOV A, H	
1E73 E6 03	8020	ANI 3	

1E75 67	8030	MOV H,A
1E76 22 FC 1C	8040	SHLD CURS
1E79 11 00 88	8060	LXI D,SCRN
1E7C 19	8070	DAD D
1E7D 7E	8080	MOV A,M
1E7E 32 FF 1C	8090	STA STR
1E81 36 FF	8100	MOV M,CS
1E83 C9	8110	RET

*INDEX BY SCREEN ADDRESS
*STORE CHAR UNDER CURSOR
*STUFF NEW CURSOR SYMBOL

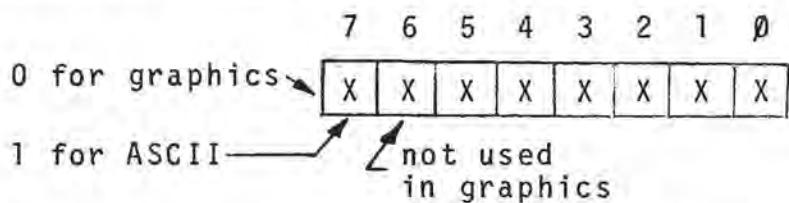
4.2 Graphics

The PolyMorphics VTI includes full graphics capability. Any or all character locations on the screen can be used in a graphics display.

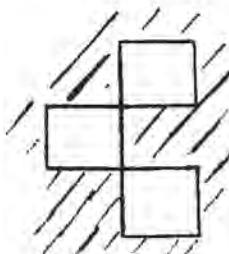
When a screen location is part of a graphics display, it is subdivided into six parts, thus:

5	2
4	1
3	Ø

(NOTE: Graphics display uses the entire screen location, including the border area that is kept dark to provide space around other characters). Each of the six "cells" of the screen location corresponds to one bit in the byte stored in the screen location. The "zero bit" corresponds to cell Ø, etc.:



Ø is "on" or "bright," 1 "off" or "dark." Thus, storing 0110101ØB (6AH) at a screen location produced this graphic at that location:



In the appendix is a chart of all 64 possible graphics characters, with their associated *nex* values

The following "game" program, called LIFE, originally invented by John Conway and popularized by Martin Gardiner in his "Mathematical Games" Section of Scientific American in 1970, illustrates the power of the graphics capability.

LIFE depicts the birth, growth, and death of a culture of cells. When a cell has one neighbor or no neighbors in the eight cells adjacent to it, it dies of loneliness. When it has four or more neighbors in the eight adjacent cells, it dies of overcrowding. It survives into the next generation whenever it has two or three neighbors. So a cell may live for just one generation, or may live for as long as the culture lives (or anything in between). A cell is born whenever an empty cell location has exactly three neighbors. (Cells are trisexual.)

The game begins with an initial entry, or Divine Creation, of a seed organism (group of cells). The initial entry can be as simple or complex as you like. The life cycle of the resulting culture arises entirely from the nature of the initial entry given the rules of LIFE.

The following program executes the rules of LIFE on the video screen in graphics. Load the program at the address indicated.

0000	0100	VADD EQU 8800H	*VIDEO BLOCK ADDRESS
0000	0110	MADD EQU 0300H	*MASTER COPY ADDRESS
0000	0120	SADD EQU 0800H	*SLAVE COPY ADDRESS
0000	0130	MAD EQU 03H	*1ST BYTE OF MADD
0000	0140	SAD EQU 08H	*1ST BYTE OF SADD
0000	0150	LINE EQU 64	*LINE LENGTH
0000	0160	TADD EQU 0208H	*TABLE (MASK & SCRATCH)
0000	0170	TAD EQU 02H	*1ST BYTE OF TADD
0000	0175	CADD EQU 0180H	*COUNT ADDRESS (GENERATIONS)
0000 21 08 02	0180	LXI H,TADD	*SET UP MASK TABLE
0003 3E 20	0185	MVI R,20H	*FIRST MASK FOR TABLE
0005 0E 08	0190	MASK MVI C,08H	*GETS EIGHT SPOTS
0007 77	0200	TABLE MOV M,R	
0008 23	0210	INX H	
0009 0D	0220	DCR C	
000A C2 07 00	0230	JNZ TABLE	*IN TABLE.
000D 0F	0240	RRC	*THEN MASK FOR NEXT LOWER BI
000E D2 05 00	0250	JNC MASK	*GETS THE NEXT EIGHT.
0011 21 00 08	0254	LXI H,SADD	*SAVE SLAVE ADDRESS
0014 E5	0256	PUSH H	*FOR USE IN LOOP
0015 21 80 01	0258	LXI H,CADD	*LOAD CADD WITH OWN
0018 36 80	0260	MVI M,80H	*SECOND BYTE TO START COUNT.
001A 21 C0 87	0270	LXI H,VADD-40H	*SET UP FOR SWAP FROM
001D 11 C0 07	0280	LXI D,SADD-40H	*SCREEN TO SLAVE WITH SLOP.
0020 7E	0282	LOOP MOV R,M	*GRAB CHAR, BEGIN MAIN LOOP
0021 2F	0284	CMA	*COMPLEMENT FOR TRUE LIFE
0022 12	0286	STAX D	*STORE ON OTHER COPY.
0023 23	0288	INX H	*NEXT
0024 13	0290	INX D	*SPOT.
0025 7C	0292	MOV R,H	*CHECK
0026 E6 07	0294	ANI 7	*LAST THREE BITS OF 1ST BYTE
0028 FE 05	0296	CPI 5	*FOR END
002A C2 20 00	0298	JNZ LOOP	*OF COPY PLUS SLOP.
002D 21 C0 07	0300	LXI H,SADD-40H	*SWAP SLAVE
0030 11 C0 02	0310	LXI D,MADD-40H	*TO MASTER
0033 7E	0312	SWAP MOV R,M	
0034 12	0314	STAX D	
0035 23	0316	INX H	
0036 13	0318	INX D	
0037 7C	0320	MOV R,H	
0038 FE 0D	0324	CPI SAD+5	*WITH SLOP
003A C2 33 00	0326	JNZ SWAP	*UP TO HERE.
003D 11 80 01	0330	LXI D,CADD	*SET UP FOR COUNT
0040 01 40 86	0340	LXI B,VADD+40H	*IN UPPER RIGHT OF SCREEN
0043 21 80 01	0350	LXI H,CADD	*WATCH THE ZERO AND CARRY!!
0046 6B	0360	MOV L,E	

0047 23	0370	COUNT INX H	*NEXT SIGNIFICANT DIGIT
0048 0B	0380	DCX B	*NEXT DOWN ON SCREEN
0049 C2 4D 00	0390	JNZ NOINC	*ZERO FLAG TO INCREMENT
004C 34	0400	INR M	
004D 1A	0410	NOINC LDAX D	*ARE WE TO END
004E BD	0420	CMP L	*OF COUNT (STORED AT CADD)?
004F DA 58 00	0430	JC OUT	*YES
0052 3E BA	0440	MVI R, 0BAH	*NO, CHECK FOR
0054 BE	0450	CMP M	*DECIMAL CARRY IN ASCII.
0055 C2 5B 00	0460	JNZ HERE	*NO
0058 3E B0	0570	OUT MVI A, 0B0H	*YES, ZERO THAT DIGIT
005A 77	0580	MOV M, A	*AND REPLACE MEMORY.
005B 7E	0590	HERE MOV A, M	*GET MEMORY
005C 02	0600	STAX B	*AND VIEW IT
005D D2 47 00	0610	JNC COUNT	*UNTIL ALL DIGITS ARE VIEWED.
0060 2B	0620	DCX H	*CHECK MOST SIGNIFICANT DIGIT
0061 BE	0630	CMP M	*AGAINST NEXT MOST.
0062 CA 67 00	0640	JZ THERE	*BOTH ZERO? EXIT.
0065 EB	0650	XCHG	*NO, INCREASE
0066 34	0660	INR M	*END OF COUNT.
0067 21 BF 02	1040	THERE LXI H, MADD-LINE-1	
006A 16 02	1050	MVI D, TAD	*GET IN POSITION FOR TABLE.
006C 01 D6 00	1060	BYTE LXI B, INST	*PSEUDO OP LIST
006F 0A	1090	BIT LDAX B	*LOAD PSEUDO OP.
0070 0F	1100	RRC	*CHECK RIGHT BIT FOR
0071 D2 87 00	1110	JNC ROT	*CELL CHECK FROM SAME BYTE
0074 0F	1120	RRC	*NO. NEXT BYTE?
0075 D2 83 00	1130	JNC ONE	*YES
0078 FE F0	1140	CPI 0F0H	*NO. ALL NGHBRS DONE THIS BYTE
007A D2 99 00	1150	JNC DONE	*YES.
007D 11 3D 00	1170	LXI D, LINE-3	*NEXT LINE ON 3X3 MATRIX
0080 19	1180	DAD D	*INCREMENT BY LINE-2
0081 16 02	1190	MVI D, TAD	*BY LINE-3+1, SINCE WE NEED
0083 23	1210	ONE INX H	*A +1 ANYWAY
0084 E6 3F	1215	ANI 3FH	*GET RID OF 2 MSB'S.
0086 07	1220	RLC	*ZERO CARRY BIT AND
0087 1F	1230	ROT RAR	*GET IN POSITION
0088 03	1240	INX B	*FOR THIS AND NEXT PSEUDO OP
0089 5F	1250	MOV E, R	*2ND BYTE FEEDS MASK TABLE
008A 1A	1260	LDAX D	*LOAD MASK FOR BIT
008B A6	1270	ANR M	*AND CHECK IT ON THE MASTER
008C CA 6F 00	1280	JZ BIT	*NO LIFE, NEXT BIT
008F EB	1290	XCHG	*BRING DOWN SCRATCH
0090 3E 07	1300	MVI R, 07H	*ADDRESS TO STORE NEIGHBOR
0092 A5	1310	ANL L	*COUNT CODED BY BIT #
0093 6F	1320	MOV L, A	
0094 34	1330	INR M	*COUNT ONE NEIGHBOR
0095 EB	1340	XCHG	*GET MASTER COPY

0096 C3 6F 00	1350 JMP BIT	*AND GET NEXT BIT IN BYTE
0099 01 BF FF	1360 DONE LXI B,0-LINE-1	*GO BACK TO BYTE
009C 09	1370 DAD B	*THAT WE'RE WORKING ON
009D 1E 00	1375 MVI E,0	*ZERO SCRATCHPAD BYTE #2
009F E3	1380 XTHL	*MOVING ON TO SLAVE COPY.
00A0 97	1390 LOAD SUB A	*ZERO A SO WE CAN
00A1 12	1400 STAX D	*ZERO NEIGHBOR COUNT
00A2 79	1410 MOV A,C	*GET INVERTED BIT MASK
00A3 0F	1420 RRC	*COMING IN BFH AND ROTATE
00A4 D2 BF 00	1430 JNC NEXT	*GOT ALL BITS?
00A7 4F	1440 MOV C,A	*NO, REPLACE MASK
00A8 1C	1450 INR E	*AND COUNT BIT NUMBER
00A9 1A	1460 LDAX D	*GET # NEIGHBORS OF THAT BIT
00AA FE 02	1470 CPI 02H	*IS IT TWO?
00AC CA A0 00	1480 JZ LOAD	*YES, CELL STAYS THE WAY IT IS
00AF 79	1490 MOV A,C	*NO, SO
00B0 A6	1510 ANA M	*KILL CELL ON
00B1 77	1520 MOV M,A	*SLAVE COPY
00B2 1A	1540 LDAX D	*HOW MANY NHBRs AGAIN?
00B3 FE 03	1550 CPI 03H	*ARE THERE THREE?
00B5 C2 A0 00	1560 JNZ LOAD	*YES, GOOD WE KILLED IT.
00B8 79	1570 MOV A,C	*OOPS, GOT TO RESURRECT IT
00B9 2F	1580 CMA	*BY INVERTING THE MASK
00BA 86	1590 ADD M	*AND ADDING
00BB 77	1610 MOV M,A	*REPLACE SLAVE
00BC C3 A0 00	1630 JMP LOAD	*UPDATE NEXT BIT IN BYTE
00BF 01 C0 FF	1640 NEXT LXI B,0-LINE	*UP ONE, WHICH IS UPPER
00C2 23	1660 INX H	*INCREMENT SLAVE ADDRESS
00C3 E3	1670 XTHL	*FOR PROPER INITIALIZATION
00C4 3E 07	1680 MVI A,MAD+04H	*END OF SCREEN?
00C6 BC	1690 CMP H	
00C7 09	1700 DAD B	*COMPLETE ONE UP
00C8 C2 6C 00	1710 JNZ BYTE	*SCREEN NOT OVER, NEXT BYTE
00CB E1	1715 POP H	*LEAVE
00CC 21 00 08	1720 LXI H,SADD	*SADD ON STACK
00CF E5	1725 PUSH H	*FOR NEXT TIME, SET UP TO
00D0 11 00 88	1740 LXI D,YADD	*SWAP SLAVE TO SCREEN
00D3 C3 20 00	1830 JMP LOOP	*ON EACH SUCCESSIVE LOOP,
00D6 C4 65	1840 INST DW 65C4H	*PSEUDO OPS CODE 48
00D8 C4 70	1850 DW 70C4H	*SPECIAL CASES: EIGHT
00DA D0 71	1860 DW 71D0H	*NEIGHBORS FOR EACH OF
00DC 87 A4	1870 DW 0A487H	*SIX CELLS PER BYTE
00DE 88 A8	1880 DW 0A888H	*RIGHT TWO BITS OF
00E0 C8 AC	1890 DW 0ACC8H	*EACH PSEUDO OP INDICATE
00E2 CC 45	1900 DW 45CCH	*WHETHER NEXT NEIGHBOR IS
00E4 84 A4	1910 DW 0A484H	*IN THE SAME BYTE AS
00E6 28 68	1920 DW 6828H	*CURRENT NEIGHBOR, OR IN
00E8 88 A8	1930 DW 0A888H	*NEXT BYTE, OR NEXT LINE

00EA C8 4C	1940 DW 4008H	*IN 3X3 MATRIX OF
00EC AC CC	1950 DW 00CACH	*NEIGHBOR BYTES
00EE 30 50	1960 DW 5030H	*NEXT THREE BITS CODE
00F0 B0 34	1970 DW 34B0H	*CELL WHOSE NEIGHBORS
00F2 54 74	1980 DW 7454H	*WE ARE COUNTING, IN
00F4 94 D4	1990 DW 0D494H	*REVERSE ORDER
00F6 58 78	2000 DW 7858H	*REMAINING THREE BITS
00F8 B8 31	2010 DW 31B8H	*CODE MASK FOR NEIGHBOR
00FA 50 34	2020 DW 3450H	*IN SAME FORMAT
00FC 54 74	2030 DW 7454H	
00FE 58 78	2040 DW 7858H	
0100 8F 2D	2050 DW 2D8FH	
0102 8C 38	2060 DW 388CH	
0104 98 39	2070 DW 3998H	
0106 FF	2080 DB 0FFH	

Screen clearing routine

ASSM(CLEAR) 0F00

0F00 21 00 88	1000 LXI H, 8800H
0F03 36 7F	1010 LOOP MVI M, 7FH
0F05 23	1020 INX H
0F06 7C	1030 MOV A, H
0F07 FE 8C	1040 CPI 8CH
0F09 C2 03 0F	1050 JNZ LOOP
0F0C 76	1060 HLT

APPENDIX A

ASCII Character set

B _i b ₇ b ₆ b ₅				0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1		
b ₄	b ₃	b ₂	b ₁	COLUMN	ROW	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	'	p	
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q	
0	0	1	0	2	STX	DC2	"	2	B	R	b	r	
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s	
0	1	0	0	4	EOT	DC4	\$	4	D	T	d	t	
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u	
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v	
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w	
1	0	0	0	8	BS	CAN	(8	H	X	h	x	
1	0	0	1	9	HT	EM)	9	I	Y	i	y	
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z	
1	0	1	1	11	VT	ESC	+	;	K	L	k	{	
1	1	0	0	12	FF	FS	,	<	L	\	l	!	
1	1	0	1	13	CR	GS	-	=	M	§	m	}	
1	1	1	0	14	SO	RS	.	>	N	~	n	~	
1	1	1	1	15	SI	US	/	?	O	—	o	DEL	

APPENDIX B Character fonts

FIGURE 15 – MCM6571A PATTERN

FIGURE 18 – MCM6574 PATTERN

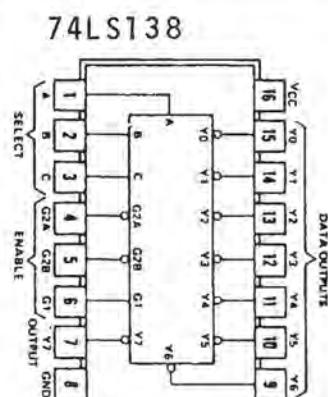
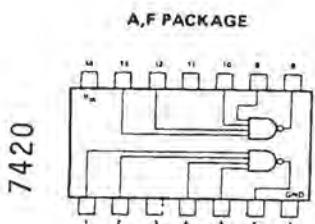
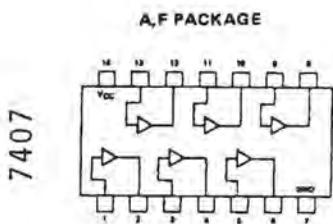
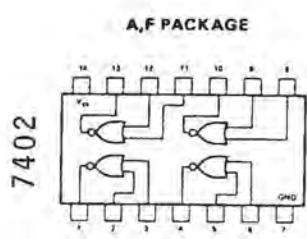
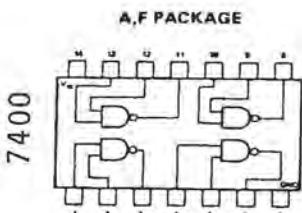
APPENDIX B Con't

FIGURE 19 – MCM6575 PATTERN

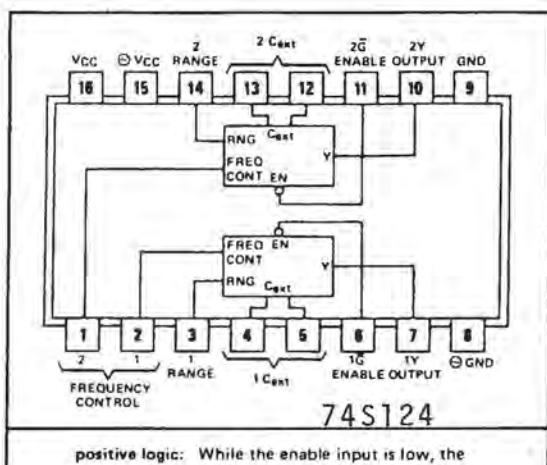
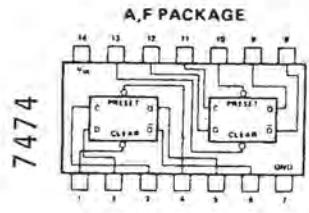
FIGURE 20 - MCM6676 PATTERN

A4	A5	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24
000	-4																
000	4																
001	-4																
001	4																
100	-4																
100	4																
101	-4																
101	4																
110	-4																
110	4																
111	-4																
111	4																

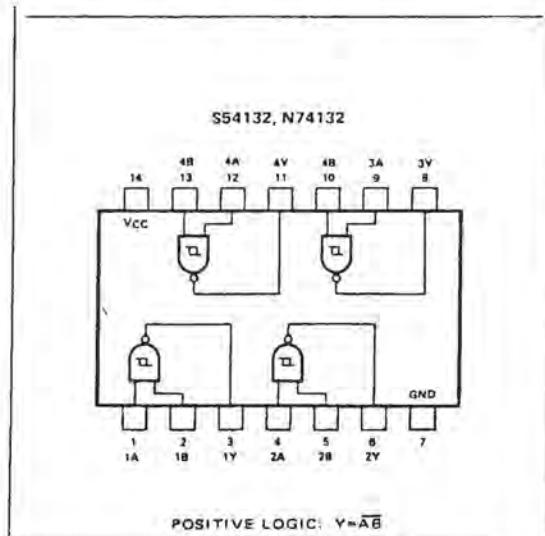
APPENDIX C Chip pinouts



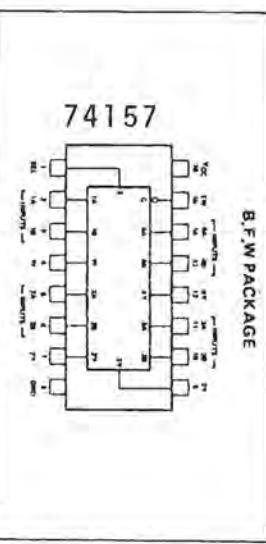
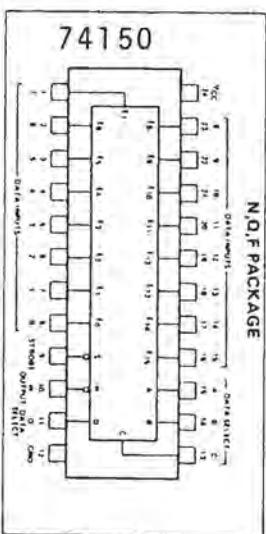
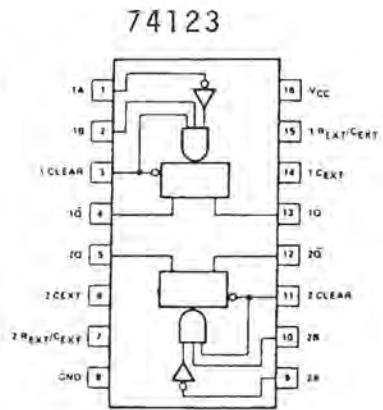
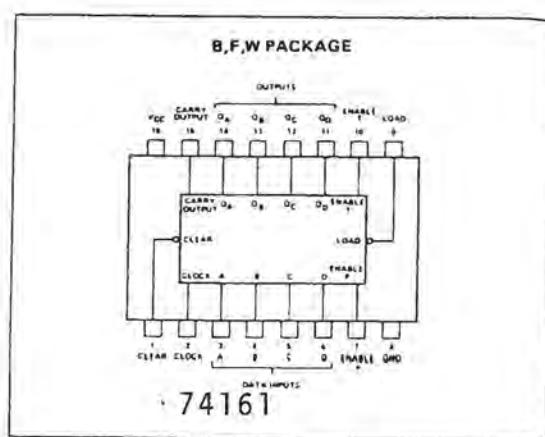
positive logic: see function table



positive logic: While the enable input is low, the output is enabled. While the enable input is high, the output is high.

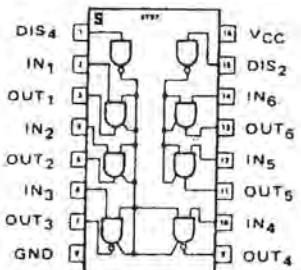


POSITIVE LOGIC: $Y = \overline{AB}$

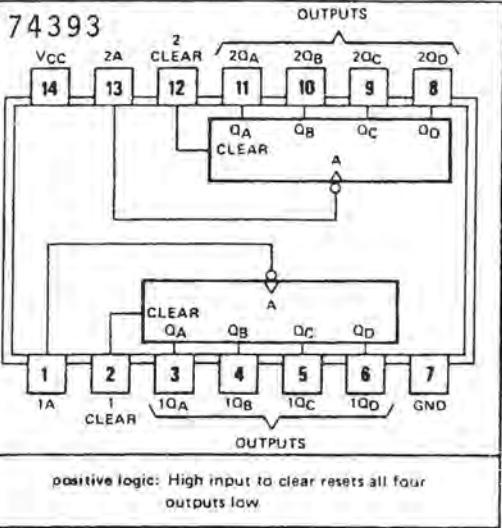


Appendix C Cont'd

8097

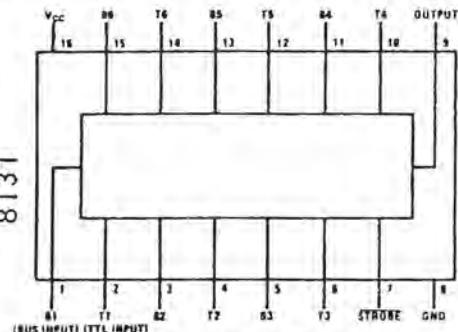


74393



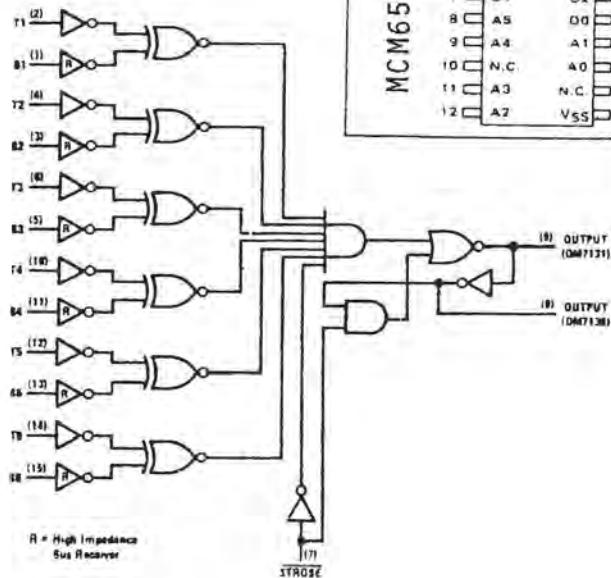
positive logic: High input to clear resets all four outputs low.

8131



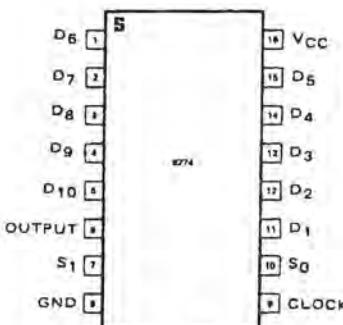
7131(J), (W); 8131(J), (N), (W);
7136(J), (W); 8136(J), (N), (W)

Logic Diagram

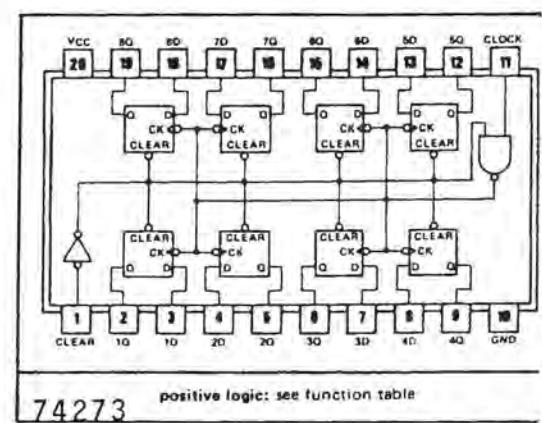


R = High Impedance
Bus Receiver

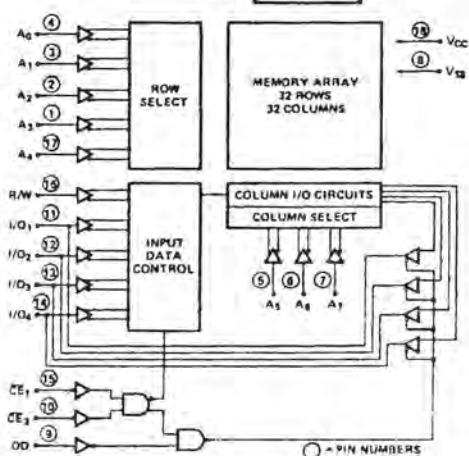
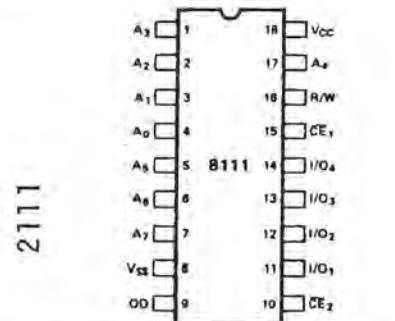
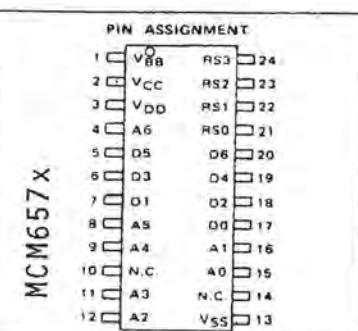
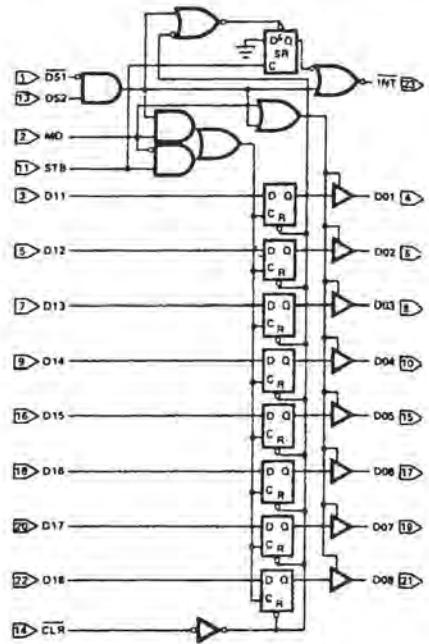
N8274B



74273

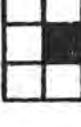
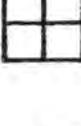
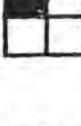


positive logic: see function table

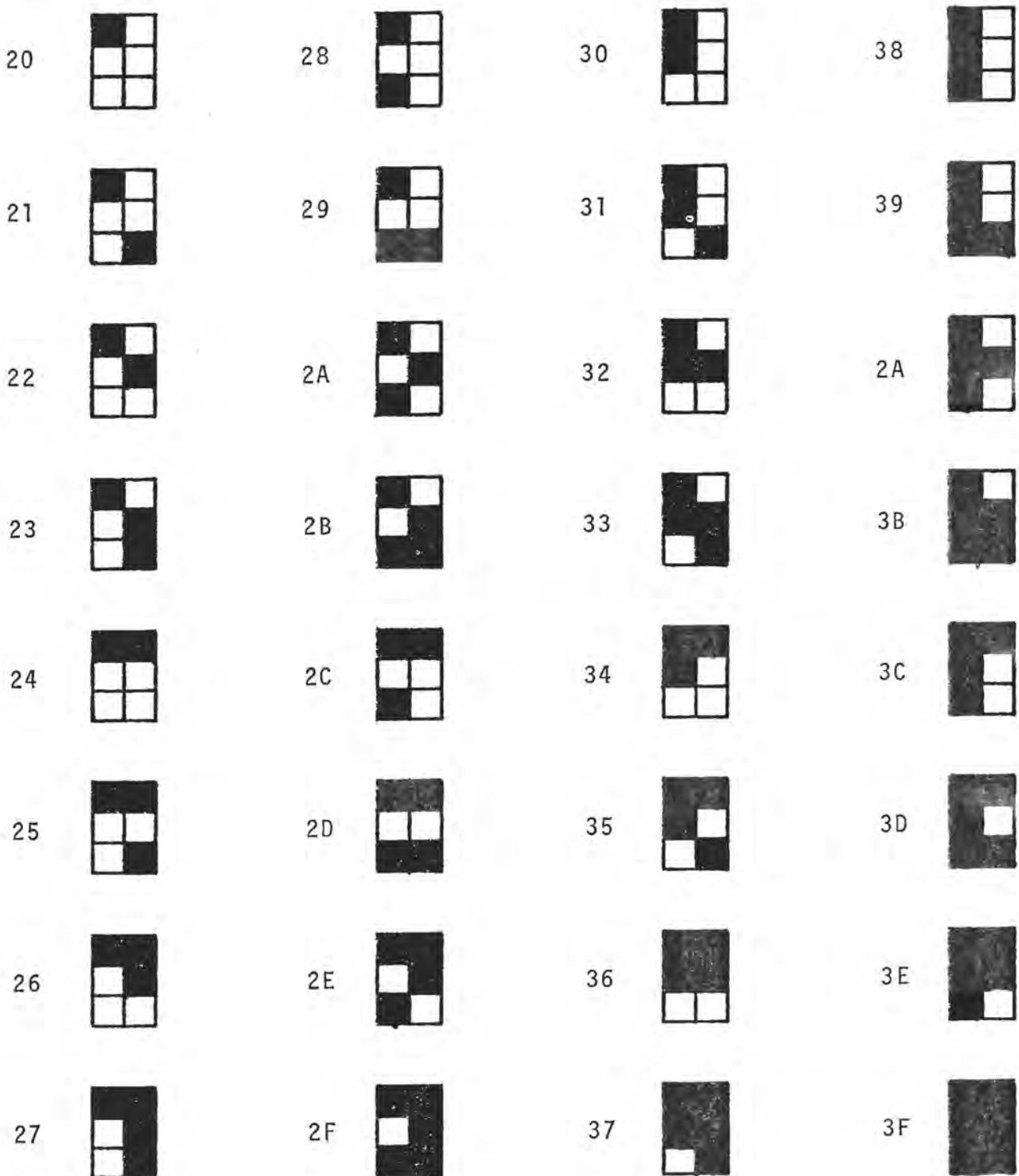


Appendix D Graphics character set

Hex Graphic (white bright, black dark)

00		08		10		18	
01		09		11		19	
02		0A		12		1A	
03		0B		13		1B	
04		0C		14		1C	
05		0D		15		1D	
06		0E		16		1E	
07		0F		17		1F	

Appendix D Con't

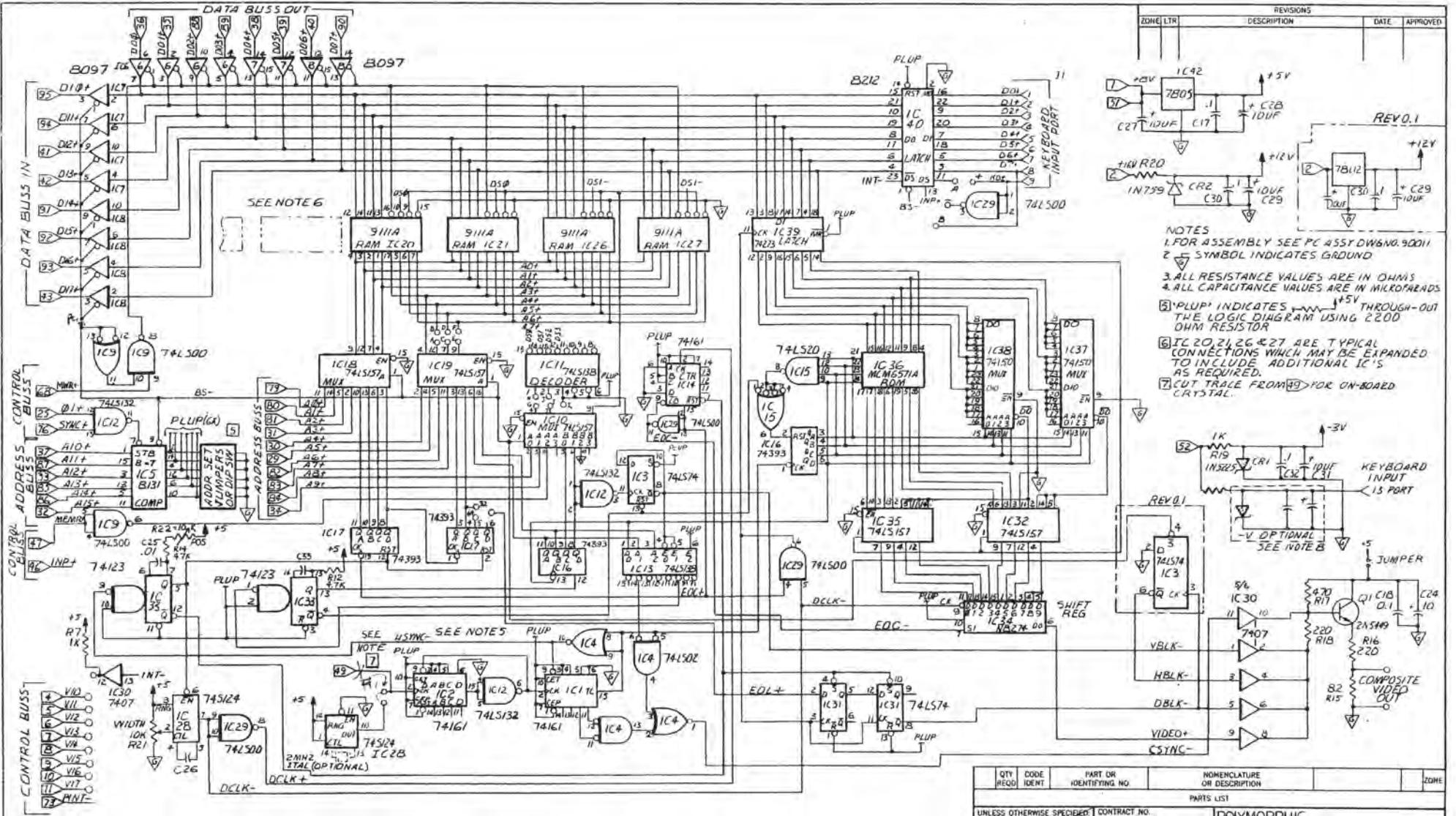


VIDEO TERMINAL SOFTWARE - COMMAND SUMMARY

	Control Character	Function
Cursor Controls	H	Home Cursor
	R	Cursor Right
	L	Cursor Left.
	U	Cursor Up
	D	Cursor Down
	E	Erase Screen
	X	Delete character
Mode Commands	I	Insert/delete mode set
	T	Text (reset I/D mode)
	F	auto line Feed mode set
	N	Normal TTY (reset ALF mode)
	S	Scroll mode set
	P	Page (reset scroll mode)

Line feed advances cursor one line, exception last line in scroll mode; then cursor fixed, and page scrolls.

Carriage return retreats cursor to beginning of line, blanking line from end unless I/D mode set.



QTY	CODE	PART OR IDENTIFYING NO.	NONCUTLATURE OR DESCRIPTION	ZONE
PARTS LIST				
UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN INCHES. TOLERANCES ON DIMENSIONS ARE .005 XXX .005 MATERIAL				
DR BY	CONTRACT NO.		POLYMORPHIC SYSTEMS ROBLES, CALIFORNIA	
CHK BY			LOGIC DIAGRAM VIDEO	
SIZE CODE/IDENT NO DWG NO D 90011				
SCALE — SHEET 1 OF 1				

NEXT ASSY USED ON
APPLICATION

DO NOT SCALE DRAWING

PLEASE NOTE

This manual has been carefully checked for accuracy, but no warranty is made as to the correctness of this document or the suitability of this product for any particular purpose. No liability is assumed for any damages, consequential or otherwise, that result from the use or misuse of this product.

WARRANTY

KIT: Defective parts will be replaced free of charge if returned to the factory within ten (10) days of receipt of delivery or upon written statement by purchaser that the unit was unassembled or untested for some longer period due to circumstances beyond his control. Completed units returned under similar circumstances will be repaired at a labor cost of \$20/hr., with defective parts replaced free.

THE WARRANTY IS VOID IF THE KIT IS SOLDERED WITH CORROSIVE FLUX.

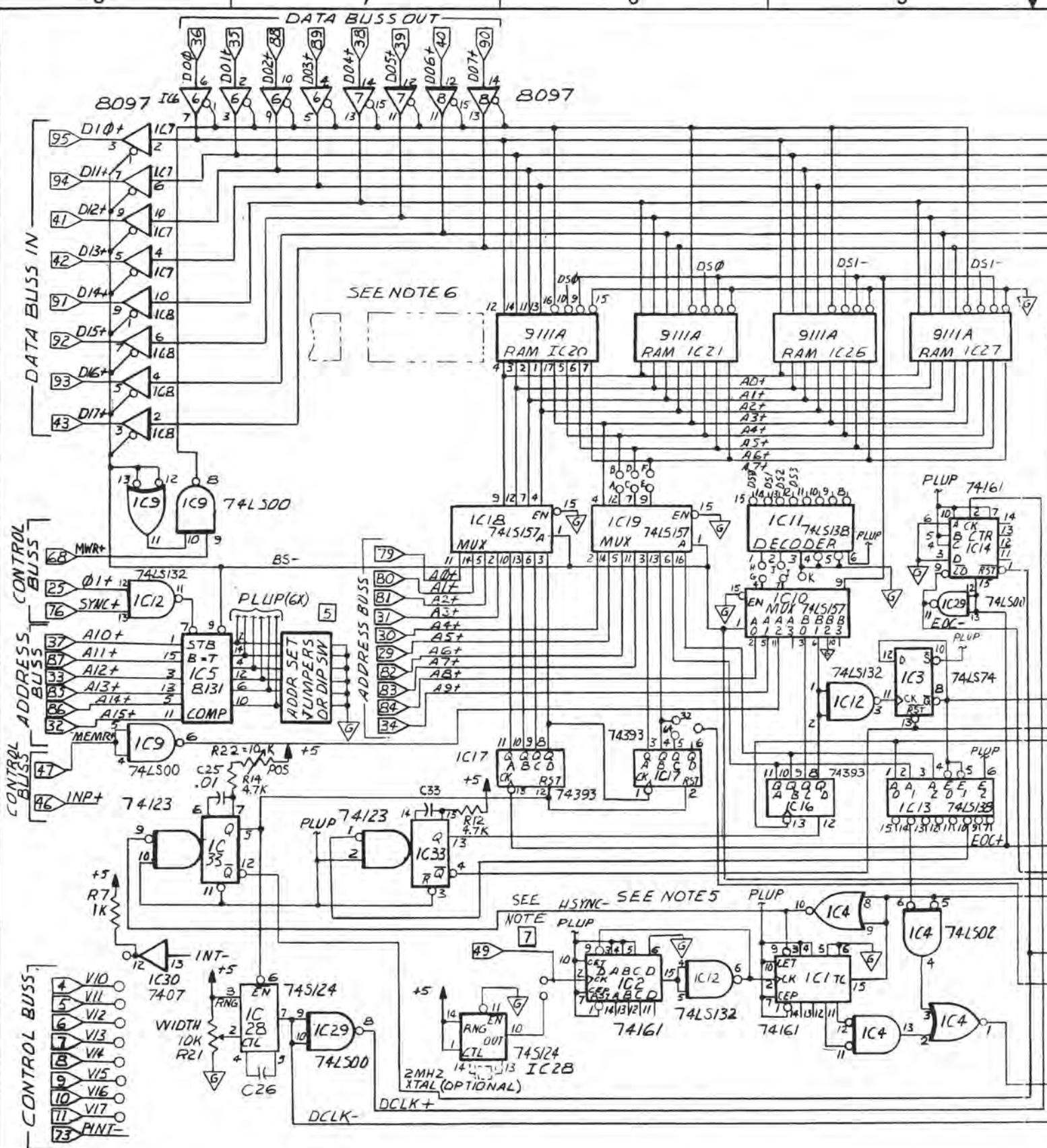
ASSEMBLED: The assembled units are fully warranted to be free of defects for ninety (90) days from the time of shipment. If they are found to be defective in this period they may be returned to the factory for repair or replacement free of charge (including return shipping).

POLYMORPHIC SYSTEMS
737 SOUTH KELLOGG AVENUE
GOLETA, CALIFORNIA 93017

LOADING DIP (DUAL IN-LINE PACKAGE) DEVICES

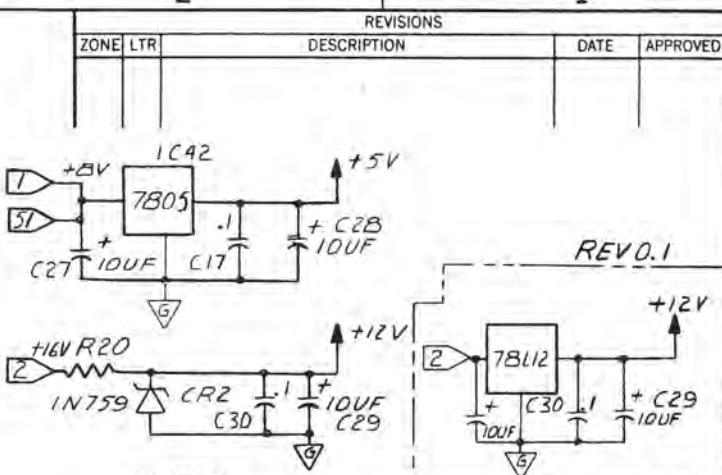
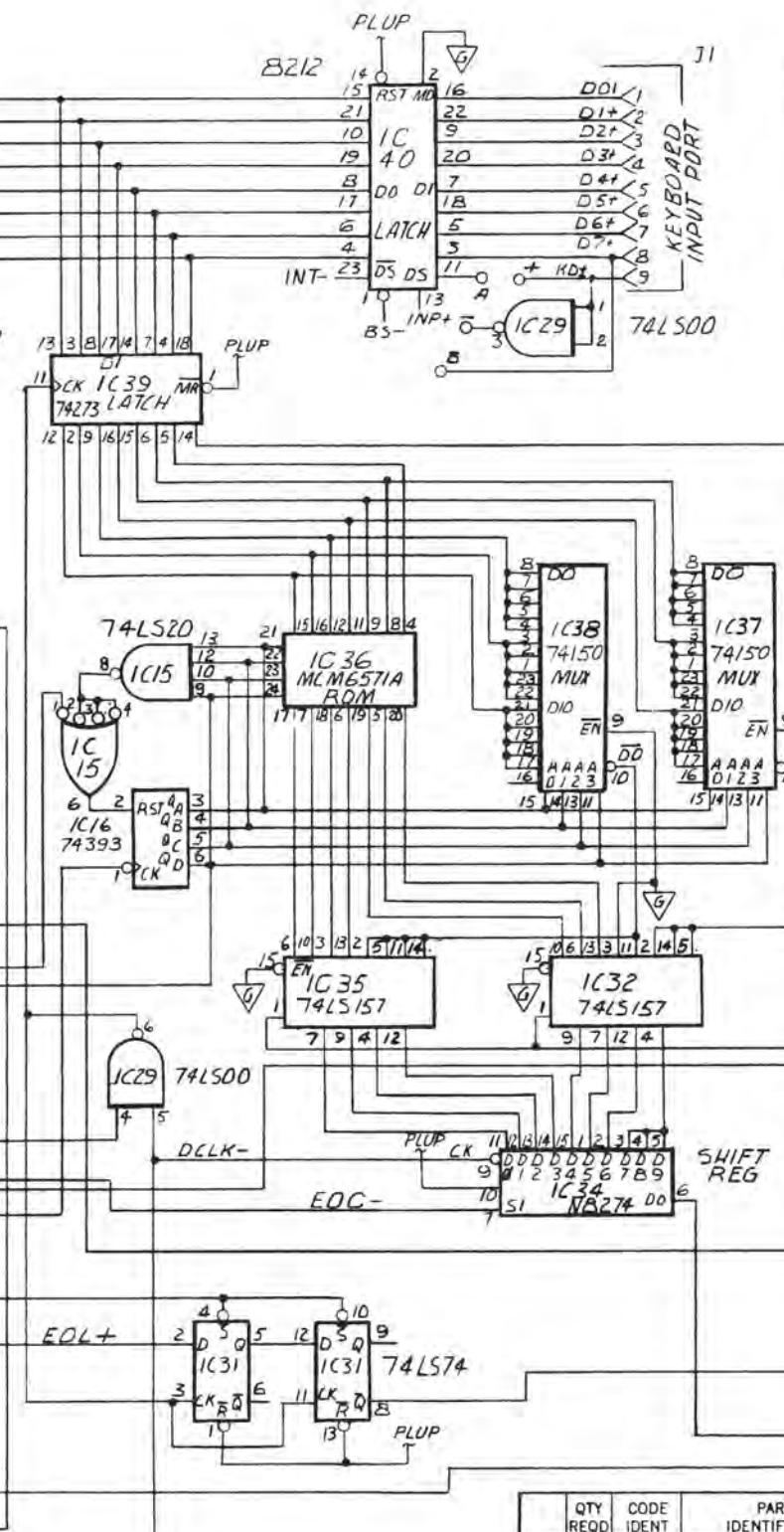
Most DIP devices have their leads spread so that they can not be dropped straight into the board. They must be "walked in" using the following procedure:

- (1) Orient the device properly. Pin 1 is indicated by a small embossed dot on the top surface of the device at one corner. Pins are numbered counterclockwise from pin 1.
- (2) Insert the pins on one side of the device into their holes on the printed circuit card. Do not press the pins all the way in, but stop when they are just starting to emerge from the opposite side of the card.
- (3) Exert a sideways pressure on the pins at the other side of the device by pressing against them where they are still wide below the bend. Bring this row of pins into alignment with its holes in the printed circuit card and insert them an equal distance, until they begin to emerge.
- (4) Press the device straight down until it seats on the points where the pins widen.
- (5) Turn the card over and select two pins at opposite corners of the device. Using a fingernail or a pair of long-nose pliers, push these pins outwards until they are bent at a 45 degree angle to the surface of the card. This will secure the device until it is soldered.



NOTES : (CONTINUED)

LAYOUT SPACE ONLY IS PROVIDED.
USED WITH USER PROVIDED PARTS
ONLY WHEN A NEGATIVE KEYBOARD
SUPPLY IS NEEDED.



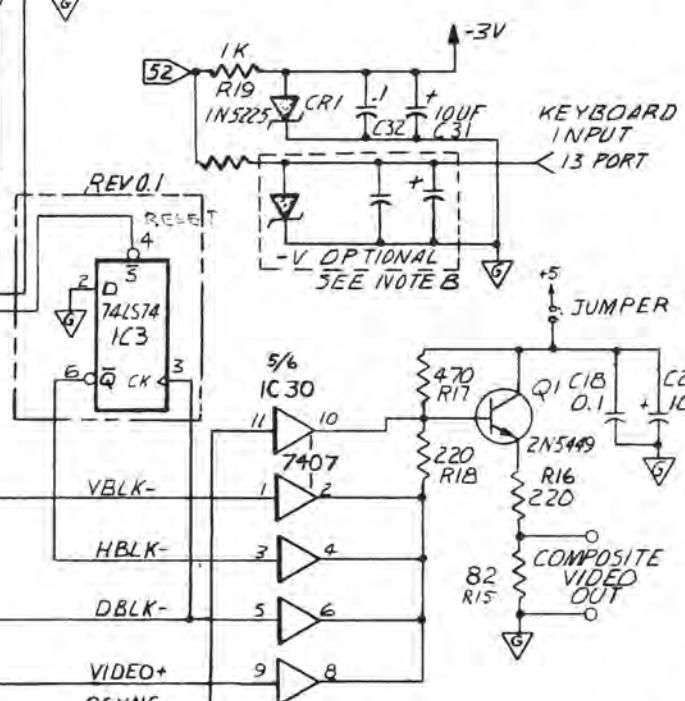
NOTES

1. FOR ASSEMBLY SEE PC ASSY DW&NO. 90011
2. SYMBOL INDICATES GROUND

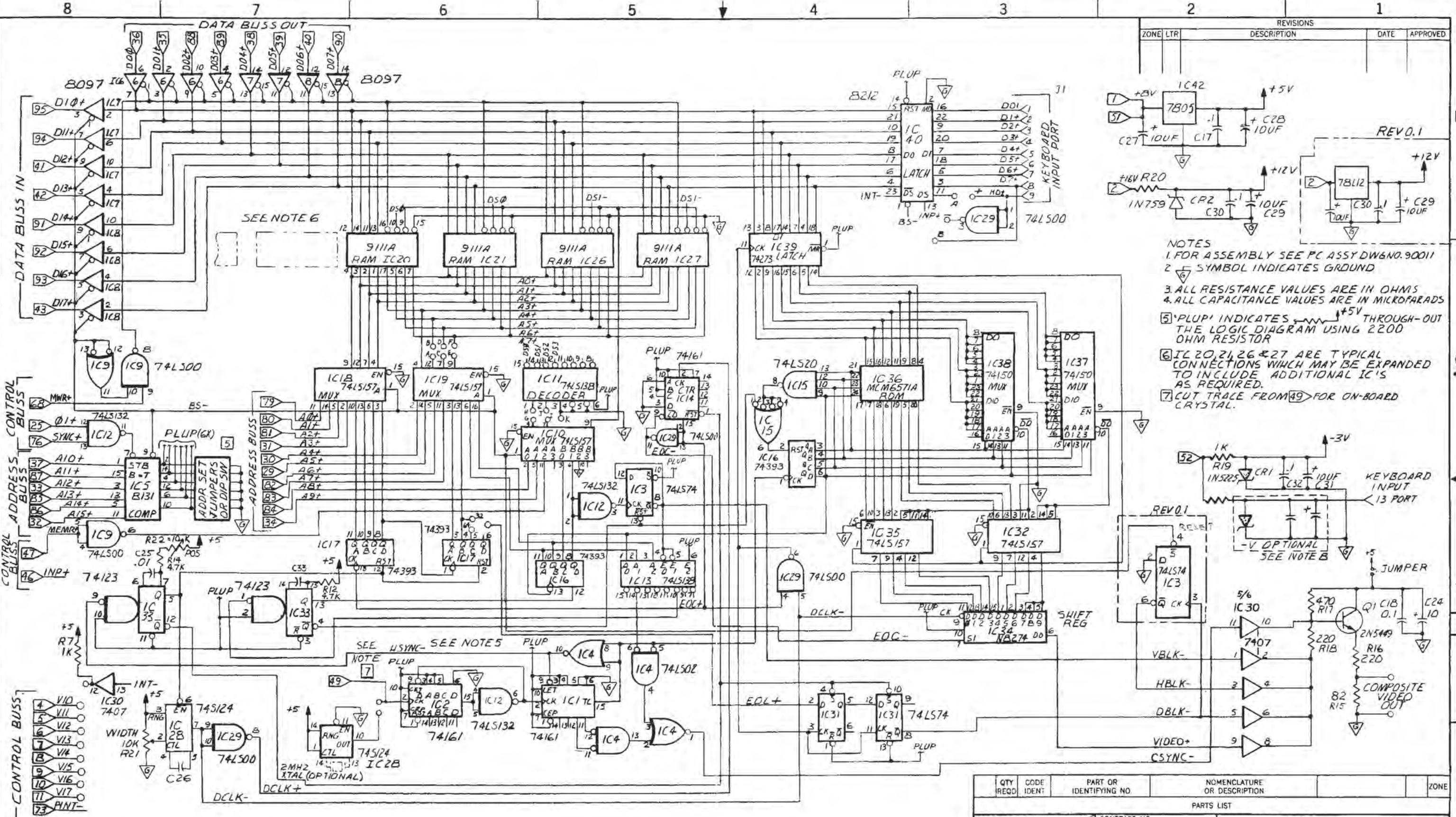
3. ALL RESISTANCE VALUES ARE IN OHMS
 4. ALL CAPACITANCE VALUES ARE IN MICROFARADS
 5. 'PLUP' INDICATES  THROUGH-OUT THE LOGIC DIAGRAM USING 2200

6 IC 20, 21, 26 & 27 ARE TYPICAL CONNECTIONS WHICH MAY BE EXPANDED TO INCLUDE ADDITIONAL IC'S AS REQUIRED.

7 CUT TRACE FROM 49 FOR ON-BOARD CRYSTAL.



QTY REQD	CODE IDENT	PART OR IDENTIFYING NO.	NOMENCLATURE OR DESCRIPTION		ZONE
PARTS LIST					
UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN INCHES. TOLERANCES ON:		CONTRACT NO.	POLYMORPHIC SYSTEMS BOLETA, CALIFORNIA		
FRAC: DECIMALS ANGLES TIONS X XX XXX ± ± ± ± ±		DR BY <u>GARLIA P.</u> 21/Mar/8	LOGIC DIAGRAM VIDEO		
MATERIAL		CHK BY	APPROVED BY		
FINISH			SIZE	CODE IDENT NO.	DWG NO.
DO NOT SCALE DRAWING			D		90011
			SCALE —		SHEET 1 OF 1



QTY IREQ	CODE IDENT	PART OR IDENTIFYING NO.	NOMENCLATURE OR DESCRIPTION	ZONE
PARTS LIST				
UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN INCHES. TOLERANCES ON: FRAC. DIMENS. ANGLES TOLERANCES X XX XXX ± ± ± ± = MATERIAL	CONTRACT NO. DR BY SARLIA P. ZIMAS CHK BY _____	POLYMORPHIC SYSTEMS GOLETA, CALIFORNIA	LOGIC DIAGRAM VIDEO	
FINISH	APPROVED BY _____			
NEXT ASSY	USED ON APPLICATION	DO NOT SCALE DRAWING	SIZE CODE IDENT NO DWG NO. D 90011	SCALE — SHEET 1 OF 1