VIDEO TERMINAL INTERFACE MANUAL

PolyMorphic Systems

Goleta California, 93017

TABLE OF CONTENTS

PolyMorphic Systems

1 Introduction

PolyMorphic Systems is pleased to have your order for POLY 88 series equipment. We have endeavored to supply the most thoroughly tested and documented material on the market. The system is modular and S-100 compatible, and is designed to accept nearly every S-100 peripheral device available. We ask you to scan this manual before assembly.

POLY 88 modules are designed for ease of assembly, use and durability. If, however, after having read the manual, you have any doubt of your faith in the project, please return the kits(s) to us in original condition for a full no-questions-asked refund.

1.1

WARRANTY

KITS: All parts and materials are warranted to be free of defects at the time of shipment. 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 up to ninety (90) days due to circumstances beyond his control. Completed units returned under similar circumstances will be repaired at a labor cost of $20/hour, with defective parts replaced free. Should the estimated cost of repair exceed 20% of the original cost of the unit, the customer will be notified prior to repair.

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).

1.2 Inspection

If your package has arrived in poor condition please inspect the contents for damage. The units are shipped in damage resistant containers. In the unlikely event of damage or breakage, please return the kit to us in the original container for replacement.

1.3 Handling Precautions:

As with any sensitive MOS (metal oxide semiconductor) caution must be exercised to avoid damage to the chip. The most frequent problem is damage caused by static electricity. While handling the chips (Integrated Circuits) we recommend that cotton clothing be worn in preference to synthetic materials.

More importantly, these devices should never be handled by the leads. They should be handled only by the ends of the chips. Since they come packed to protect the leads, there is no reason to actually endanger the chip until it is time to install them in the IC sockets on the board.

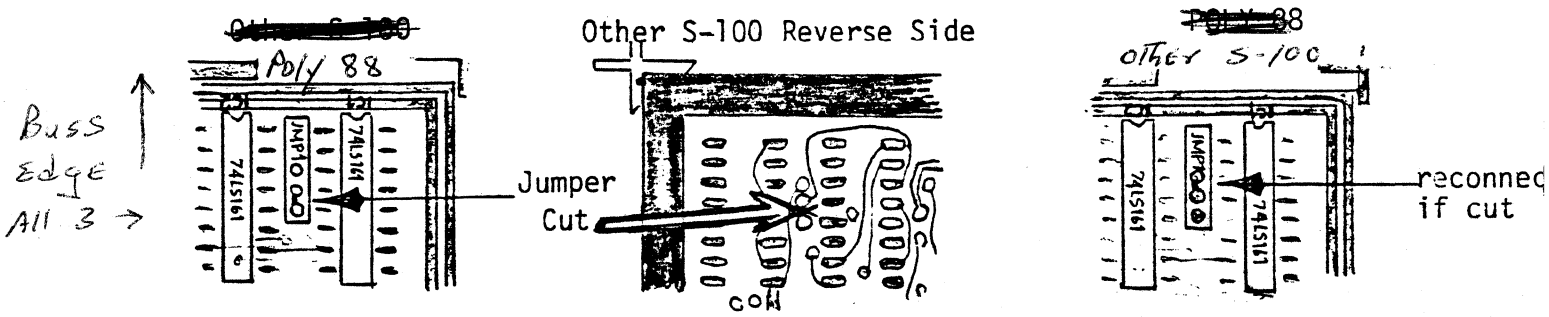
1.4 Soldering Tips:

1. Use a soldering iron of 25 watts or less. Larger soldering tools such as soldering guns and bigger irons are too hot. The lower wattage irons do the job efficiently and reduce the risk of burning the printed-circuit board.
2. Use a small, clean tip on the iron. Clean it after each use on a small piece of damp sponge.
3. Use the 60-40 rosin-core solder. This type is provided with your kit. Use the supplied solder or the smallest diameter available. Do not use acid-core solder or externally applied fluxes. <u>USE OF EXTERNAL FLUXES OR ACID CORE SOLDER VOIDS YOUR WARRANTY</u>.
4. To solder, first apply a light coat of solder to the tip of your iron. Place the tip against both the component lead and printed circuit juncture to be soldered. Add ample solder to the juncture of lead and printed circuit pad but not to the iron itself. The solder will melt when the unit to be soldered is sufficiently heated and will bond by forming a capillary film between the lead and pad.
5. Remove the solder after one or two seconds. The rosin will bubble (boil) out. Allow three to four bubbles to form before removing the iron. <u>Do not keep the heat applied for more than ten seconds</u>.

4.0 Option Selection

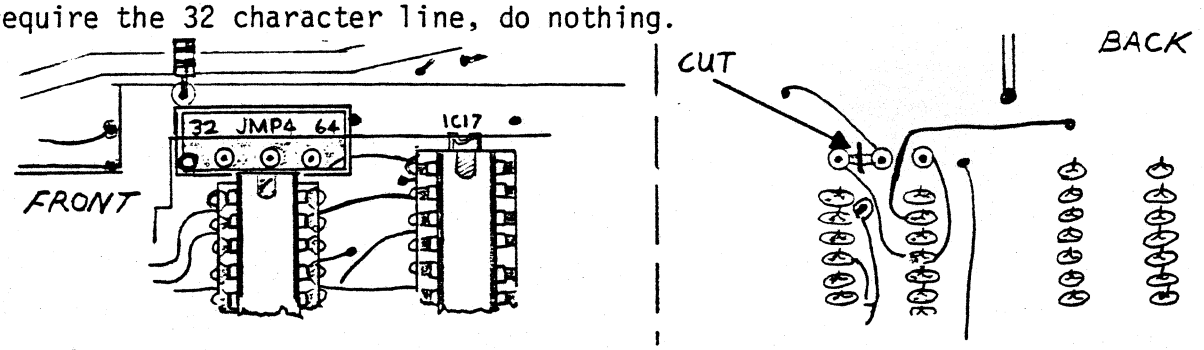
Though the VTI is an integral part of the POLY 88 system, it is compatible with other systems. JMP 1 changes the divide ratio from the system clock to produce scan rates which are more appropriate when using different system clock rates.

No change should be made if the VTI is to be used with a POLY 88. For other S-100 type systems a jumper should be cut, as noted in the drawing below and the designated jumper should be added as shown. Should you wish to use the VTI in a POLY 88, simply re-jumper at JMP 1 as shown.



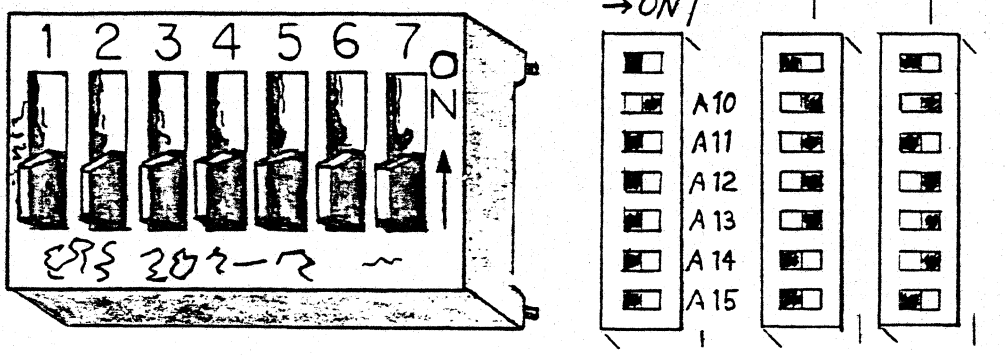
4.1 Select Character line length option.

Your board is configured for a 64 character line. If you require the 32 character line, cut the trace on the back of the board between the middle pad of JMP4 and the pad designated 64 at JMP4. Install a jumper between the middle pad of JMP4 and the pad marked "32". If you do not require the 32 character line, do nothing.



4.2 Address location:

The VTI interacts through the S-100 bus as a block of memory and input port for keyboard. The memory block, (½ or 1 K bytes, depending on option) can be located at any address from Ø through 63 K in 1 K increments. Software written for this product will usually locate it at hexadecimal address 88ØØ in systems other than the POLY 88, where it is at F8ØØ. Set the address to 88ØØ or F8ØØ as required by matchine the appropriate figure on the next page.



black is switch

4.3 Interface TV monitor or TV receiver:

At this point, your unit should operate if connected via coaxial cable to either video monitor or slightly modified receiver. (For the Hitachi line, an inexpensive TV receiver modification kit is available through PolyMorphic Systems - order P/N 100011).

Because of rigid FCC regulations, the circuit has been designed for direct connection to the video input circuit of the video amplifier, which is located between the last video IF stage and the video output circuit.

When the circuit is broken at video amplifier input, a DC bias circuit for the stage will probably be necessary, since in most cases it is supplied from the video IF amplifier. The optimum interface circuit will vary, but frequently a capacitive coupling to a resistive bias circuit is adequate. The coupling capacitor is typically a 1-5 $_{u}$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.

4.4 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 interfaces to any ASCII keyboard. 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 strobe on your keyboard is "positive going" (rising in voltage to indicate keystrike) or "negative going" (dropping to indicate keystrike). If you use the PolyMorphic Systems keyboard the proper options are already prewired on the board go to section 5 for checkout.

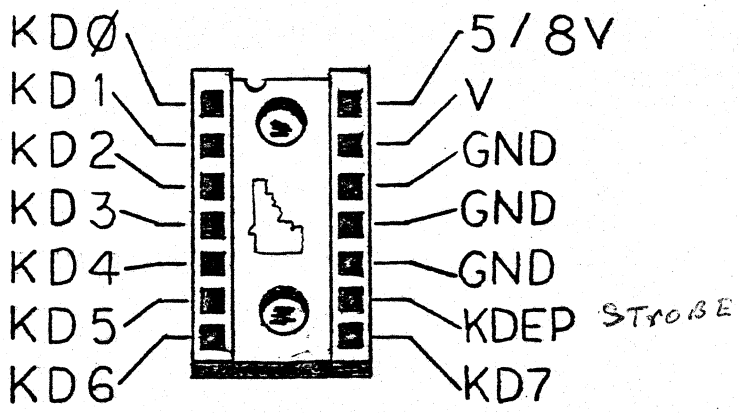
4.4.1 Connector configuration

The parallel input from the keyboard is designed to come in over a ribbon cable terminated by a DIP MALE CONNECTOR. This plugs into the 14 pin DIP socket at the upper right hand corner of the board. The 8 parallel input lines are connected to pins 1 through 8 of this socket (J-1) with 1 being the least significant bit. Pin 9 carries the "positive going" or "negative going," strobe. Pins 10, 11, and 12 are grounded. Pin 13 is the output from the optional *negative voltage regulator. Pin 14 carries +5 volts as the primary supply for most keyboards. JMP8 allows 8 volts unregulated power at Pin 14 if desired. Be sure to cut the trace connecting 5 volts if you require this option.

* Used when the keyboard requires a negative supply. The user should select and obtain the components suited to his keyboard. See section 4.5.

A jumper is inserted from the middle pad of JMP8 to the pad nearest the regulator within the area designated JMP8. See Appendix for Jumper instructions.

WARNING: FAILURE TO CUT THE TRACE SUPPLYING 5 VOLTS WHILE ATTEMPTING TO JUMPER IN 8 VOLTS WILL DESTROY EVERY COMPONENT ON THE BOARD AND VOID THE WARRANTY!



4.4.2 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. Proper use of the keyboard requires that the processor must establish two conditions before using the input data. It must indicate 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 parallel with the data lines. This line signals each depression by a pulse. This test-function informs the processor that the necessary input conditions have been met. The pulse:

1) interrupts the processor by setting an interrupt service latch contained on the input buffer, or

2) the interrupt request latch is available on data bit Ø of the status port; the keyboard strobe is available on data bit 7.

4.4.3 Keystrobe Seclection

The Keydepressed strobe may be one of four types. Attach a strobe line to a logic probe to determine the type:

1. It may be normally low, (below 0.8V) go high (above 2V) when a key is depressed, and return low when it is released.

2. The keystrobe may be normally high, go low on a key depression, and return high on release.

3. The keystrobe may be normally low, generate a positive pulse on key depression and immediately return low.

4. It may be high and generate a negative going pulse on key depression.

If you keyboard is type 1 or 3, the jumper is already configured correctly.

If it is a type 2 or 4, cut the minus trace from the center pad of JMP7 and jumper from center pad to + labeled pad.

4.5 Optional voltage regulator

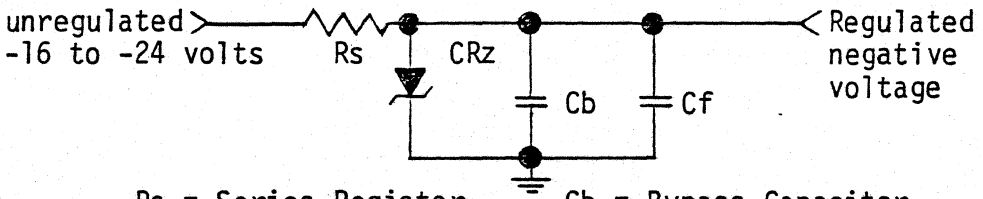
Provision has been made for the optional negative voltage regulator required by a number of keyboards. The pads and traces for this voltage supply are located adjacent to the keyboard input socket, just above the IC23. The supply regulate the -16V line by means of a resistor and zener diode stabilized by two capacitors. The four components are R14, C29, C28 and D2. The choice of resistor and zener values depends on the voltage and current requirements of the keyboard.

4.5.1 Installing Optional Voltage Regulator

The component values of the customer provided zener keyboard supply must be calculated. The values depend not only on the required voltage, but also the required current.

The required voltage and current must be obtained from the keyboard manufacturer or distributor.

The supply curcuit is represented by the following schmatic (the component labels have been generalized to avoid conflicts between different board revisions):



Rs = Series Registor Cb = Bypass Capacitor

CRz = Zener Diode Cf = Filter Capacitor

The bypass capacitor (Cb) should be a 0.1_u F or 0.01_u F ceramic disc; the value is not critical. The filter capacitor (Cf) should be a 10_u F 25-35 volt tantalum with the positive lead to ground (ground is positive with respect to the negative regulated voltage).

The series resistor (Rs) and zener diode (CRz) are more difficult to calculate. There are two values that must be calculated for each part -- resistance and wattage for Rz, voltage and wattage for CRz.

1. CRz Voltage; should have voltage equal to the required regulated voltage.

2. Rs resistance; to determine the resistance of Rs, use the specified unregulated voltage value closest to zero. This is -16 volts according to bus specifications. Take the difference between this value and the regulated value.

EXAMPLE: for regulated -12 volts, -12-(-16) = 4 volts. Divide the remainder by the maximum required current in amps.

EXAMPLE: for 10ma current = 0.010 amps, 4 volts/0.010 amps = 400 ohms.

Use a convenient standard resistance approximately 20 percent lower than the calculated value.

EXAMPLE: 440 ohms -20 percent = 400-80=320, 320 ohms is not a standard value, use 330 ohms or 270 ohms.

3. CRz wattage. To determine the wattage rating for CRz use the

worstcase current assuming all the current passes through the zener (this can happen if the keyboard is disconnected and the -16 supply is unloaded).

EXAMPLE: Using Rs=330 ohms, Iwc = 12/330 ohms - 0.03636 amps. Now calculate the wattage for CRz.

EXAMPLE: 12 volts x 0.03535 amps = 0.436 watts. Use a higher wattage than calculated, like ½ watt or higher for the given example.

4. Rs wattage. Use the worst-case current determined in calculations for CRz wattage (Iwc) and calculate the required wattage.

EXAMPLE: $Prs = (Iwc)^2 \times Rs = (0.03636)^2 \times 330$ ohms = 0.436 watts. Use the next highest standard value, like ½ watt for the given example.

Install the components (note the capacitors Cf and Cb can be in either capacitor position -- they are in parallel -- as long as the tantalum polarity is correct).

5. VTI Checkout

Install the VTI in your system and connect a video monitor or modified TV set to the video out connector. A keyboard is not needed at this time.

Check the following points for voltages within the ranges indicated.

( )	IC37	Pin 2	+4.75 to + 5.25V
( )	IC37	Pin 3	+11.4 to +12.6 V
( )	IC37	Pin 1	-2.2 to - 3.3 V

If these voltages are not correct, check all IC's for proper case temperature. If any of the DIP packaged IC's are running hotter than 90°C (195°F) (i.e. - if you can't hold you finger on them) they should be removed and the voltages re-checked. If the voltages are now normal you have found a bad IC.

If you find bad components on the VTI, and it is in warranty, return the defective part to PolyMorphic Systems and a new one will be mailed to you at no charge. Include with the part a note explaining the problem and the serial number of your Poly 88 or the sales order number off the packing slip.

If voltages are still not normal proceed to the troubleshooting section.

Enter one of the following two programs into your computer. Note that one is assembled for Ø and one for ØC8ØH. Use the program for which you have RAM available (ØC8ØH for Poly 88's).

TEST PROGRAM 1

Address	Data		Program	
Ø	21		LXI	H,88ØØH
1	ØØ			
2	88			
3	75	LOOP:	MOV	M,L
4	23		INX	H
5	7C		MOV	A,H
6	FE		CPI	Ø8CH
7	8C			
8	C2		JNZ	LOOP
9	Ø3			
A	ØØ			
B	76	WAIT:	HLT	
C	C3		JMP	WAIT
D	ØB			
E	ØØ			

TEST PROGRAM 2

Address	Data		Program	
ØC8Ø	21		LXI	H,ØF8ØØH
ØC81	ØØ			
ØC82	F8			
ØC83	75	LOOP:	MOV	M,L
ØC84	23		INX	H
ØC85	7C		MOV	A,H
ØC86	FE		CPI	ØFCH
ØC87	FC			
ØC88	C2		JNZ	LOOP
ØC89	83			
ØC8A	ØC			
ØC8B	76	WAIT:	HLT	
ØC8C	C3		JMP	WAIT
ØC8D	8B			
ØC8E	ØC			

Run the program at Ø (program 1) or ØC8ØH (program 2) following the instructions provided with your computer. The programs should produce a display of the ASCII character set and graphics characters on your TV screen. Adjust the horizontal and vertical hold controls for a stationary display.

If you cannot get a stable display check the connections to the RV for continuity and to make sure that the signal and ground leads are not reversed. Refer to the troubleshooting section if you cannot get a display.

Potentiometer R28 controls the position of the left-hand edge of the display. R27 controls the width of the display. Adjust R27 and R28 for proper position and width of the display on your TV screen. The controls interact slightly so 2 or 3 iterations may be required.

The height of the display is not adjustable on the VTI board (it is set to EIA standards). In some cases portions of the top or bottom line of the display may be off the edges of the screen. The height may be adjusted by the "vertical height" and vertical linearity" controls on the back of the TV. These are usually screwdriver adjustments and in some cases the rear cover may have to be removed to access them.

Use only an <u>insulated</u> screwdriver, or other alignment tool, for adjustment. On some sets the screw adjustments may have voltages on them. Adjust both the linearity and height to bring all 16 lines onto the screen. These controls interact heavily and there will be several combinations which will bring the display onto the screen. They should be adjusted such that the display is linear - the characters in the first and last lines are the same height.

Turn off your system, and attach a properly wired keyboard to the keyboard socket (J1). Enter one of the 2 following programs into your computer and run one at the address indicated.

Address	Data	Program		
Ø	F3		DI	
1	21		LXI	H,Ø88ØØH
2	ØØ			
3	88			
4	ØC	LOOP:	INR	C
5	C2		JNZ	LOOP
6	Ø4			
7	ØØ			
8	DB		IN	Ø89H
9	89			
A	E6		ANI	1
B	Ø1			
C	C2		JNZ	LOOP
D	Ø4			
E	ØØ			
F	DB		IN	Ø88H
1Ø	88			
11	F6		ORI	8ØH
12	8Ø			
13	77		MOV	M,A
14	23		INX	H
15	C3		JMP	LOOP
16	Ø4			
17	ØØ			

Address	Data		Program	
ØC8Ø	F3		DI	
ØC81	21		LXI	H,ØF8ØØH
ØC82	ØØ			
ØC83	F8			
ØC84	ØC	LOOP:	INR	C
ØC85	C2		JNZ	LOOP
ØC86	84			
ØC87	ØC			
ØC88	DB		IN	ØF9H
ØC89	F9			
ØC8A	E6		ANI	1
ØC8B	Ø1			
ØC8C	C2		JNZ	LOOP
ØC8D	84			
ØC8E	ØC			
ØC8F	DB		IN	ØF8H
ØC9Ø	F8			
ØC91	F6		ORI	8ØH
ØC92	8Ø			
ØC93	77		MOV	M,A
ØC94	23		INX	H
ØC95	C3		JMP	LOOP
ØC96	84			
ØC97	ØC			

These programs will take a character from the keyboard when a key is depressed and display it on the screen. The display position will advance 1 character position everytime a key is depressed. Carraige returns and line feeds are not recognized as such and will appear on the screen as Greek letters or special symbols. On 32 character boards the first 32 characters in each line are displayed and the second 32 will not be displayed. When running the normal video driver a carriage return moves the cursor to the beginning of the next line.

If this test does not work, check the keystrobe polarity and your keyboard wiring, otherwise see the troubleshooting section.

This completes the setup of your PolyMorphic Video Terminal Interface. See section 7 for operation of the software supplied with the VTI.

5.1 Troubleshooting the VTI

In addition to a video monitor, you will need a simple logic probe with pulse detector. If you do not have one, buy one or build one using the circuit enclosed. If you cannot use a logic probe, do not attempt detailed checkout.

You will also need the VTVM or VOM you used earlier. A magnifying glass will also be helpful.

If the system does not operate properly, first eliminate the most common problems:

( ) 1. Check the components on the board for proper location and orientation. In particular, check the tantalum capacitor orientation carefully.

( ) 2. Check the board to make sure there are no solder bridges.

( ) 3. Check that all jumpers are in place, and that they are correct for either the 32 or 64 character option, whichever you ordered.

( ) 4. Check all boards to make sure that all IC pins are correctly inserted -- not folded under or broken off, etc.

( ) 5. Check the jumpers or DIP switch on the video board for proper address selection.

If these problems are eliminated, and the system still does not run properly, check the CPU board, using the logic probe pulse detector, to ensure that the clock signal is available on pin 49 of the edge connector.*

* Pin 49 is the second pin from the right on the top of the bus (49th from left).

Load test program 1 or 2 and run it. The character set should appear on the screen.

The video board consists in essence of three areas: Sync, Data Bus, and Character Generation-Video.

If you have a coherent, stable, but useless display, the problem is most likely in Data Bus.

If you have no display, or all graphics, the problem is most likely in Character Generation-Video.

One problem that affects all three areas is the output buffer, so begin by checking pinouts on:

( ) IC 31 (out buffer, 7407).

Next, perform the relevant steps below:

Data Bus

( ) Check all RAMs, ICs 21 through 28 (91L11 or 2111).

( ) Check all RAM pins for proper insertion.

( ) Check for solder bridges on RAMs and in the bus driver area.

Character Generation-Video

( ) Check the dot clock chip, IC 29(74S124). If you have a display, you can check IC29 by decreasing the display width by adjusting potentiometer R27. If the display width changes evenly, the dot clock chip is probably good.

( ) Check the shift register, IC 35 (8274).

If you have done all the above, and still have system malfunctions, continue with detailed checkout below. 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 electrical test patterns, work from end results backward 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 contriubtions to character presentation.

5.1.1 Power Mains

If visual inspection looks good, see if the power mains are proper. There should be +5.0 $\pm$ 0.25 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 right 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, especially along the top of the board.)

B. 5 volt bus: the bottom lead of resistor R12. 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(6571-4). Measure +12V$\pm$ 5% at the junction of R20/C29.

D. VBB bus for IC37: Measure -3V $\pm$ 10% at the left hand lead of D1. (This is the only negative voltage.)

5.1.2 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.

5.2 Signal tracing

Unsolder the right end of the 100 ohm R1 (junction with pins 2, 4, 6, of IC31 ---- 7407) and attach a clip lead to the free end of the resistor for use as a scope probe. (Keeping a wire in the hole 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 IC31 (7407) 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.

5.2.1 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.

5.2.2 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 what is missing. If, in the "normal" connection (i.e.: lead clipped to where 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 de-

rived from that clock. (The board is testable with nothing more than proper power supplies and a clock for inputs.)

No video pattern? Let us see if it is shifting out of the register IC35-6(8274) (pin 6 of IC35). Got it? Then the path through IC31 is not passing it. Check for it at the input pin 9 and output pin 8 of IC31. 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 IC35-6? 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?

First the dot clock on IC35-9(8274): 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 bankwidth 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 on IC30-8(74LS00). Absence of sync should not prevent this display from being recognizable.

EOC (end-of-character) loading signals on IC35-7 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 IC35-9(8274) 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 R19, 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 IC33 and IC36(74LS157). Also the outputs ROM IC37 (6571) and the graphics generators IC38 and IC39 (74150).

The patterns associated with outputs from IC38 and IC39 (74150) have a right to change every 5 sweeps. At the IC39 (74273) inputs to the display generators IC37, 38, and 39, (6571) 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 nothing that the fineness of detail is less at the input to generators than at the output is usually sufficient for trouble localizing.

The screen pattern for any significant bit input to the generators should be traceable back through corresponding pins of the sampling latch IC40 (74273) to the same significant bit of the internal data bus. But remember, the nth character in memory 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 trouble-shooting without sync, anyway.

In like fashion, grounding pin one of IC33(7LS157) forces MUX's IC33 and IC36 (74157) to select only graphic symbols from IC38 and IC39 (74150). 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 IC33-1(74157) will show which memory locations contain graphics or non-graphics characters. An MSB in memory is inverted in the latch to select graphics.

5.2.3 Localizing on the EOC (end of character) path

If you had dot clock input to shift register IC35-9(8274) but no strobe (IC35-7) 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 IC30 (74LS00) 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 IC13-2. 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. IC13-14 (74161) 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 IC13-15 is the EOC (end of character) signal. It should appear at the input to the symbol counter IC16-13. An inverse (negative true) of this pattern should be found as loading signals n latch IC40-11 (74273) and shift register IC35-9. 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.

5.2.4 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 IC30(74LS00), 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 IC29-7 (745124). Using a voltmeter, check its "width" pot for the ability to vary IC29-2 from zero to 5 volts. Check also for the enabling portion of the horizontal blanking signal on IC29-6. 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 IC29-6 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 IC29-6. The value is under control of the "pos" pot which varies the time delay (and thus the average DC value) the the blanking monostable.

5.2.5 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 R19 does not open the composite sync path at IC31-10 (7407). 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 IC29-6 (74S124). 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 on time calibration) but, if you can

change it, it is working. Perhaps easier to see is its inverse - a logic high on IC34-5 (74123). 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, $\bar{Q}$ 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 IC34 monostable multi-vibrator to stretch the sync into a wider blanking? The carry-out of counter IC1-15 (74161) should have its inverse on IC34-9 (74123). This is a 4½ microsecond pulse every 58½ microseconds.

Actual horizontal sync is the same width, but 4½ microseconds later, and can be seen on IC3-13 (74LS02). Its inverse is one IC3-1 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 5.3.1.

If these are NOGO, is the system clock on edge pin 49 and is it reaching IC2-1 (74161)?

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 5.2.3.)

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.

IC2-2 has an inverted form of IC1-15 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 IC2-11 (74LS138), IC2-13 (74161), IC2-15, or IC3-13 (74LS138) have an observable once-per-sweep bar, horizontal sync seems to be doing its job.

5.2.6 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 IC15-1 (74393) is a once-per-sweep pulse which may not be in the visible portion 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 IC15-2. The reset is like the clock on IC15-1 except a) it occurs every 15th sweep; b) it is a 4½ microsecond darkening instead of a brightening; and c) it occurs 4½ 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 IC15 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 IC16-3 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 IC15-13. 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, IC15-11 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 IC15-8, IC2-9 shows the bright inverse of 8 sweeps of vertical blanking at 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 IC16 (74394) can be directly inferred from the theory discussion and the pin outs of the 74393. The EOC pulses described in a.2.3 are seen as a vertical bar per symbol space on IC16-13. 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 IC34-5. (Refer to Section a.2.5.)

The functions of IC12 (74138) and IC34 (74123) are not directly observable in the presence of sync. If no sync at all is reaching the raster, normal operation of IC34-13 can be noted as small (on the order of 30 nanoseconds) 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 IC34-1 is held to ground.

The combination of IC34b and IC12 can be checked by grounding pins 4 and 5 of IC3. 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 IC12-12 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 ÷ 2 relationships for ascending orders of bits. These patterns should be traceable through the MUX's IC's 17, 18, and 19 (74157) and decoder IC11 to the associated RAM address input pins.

Normal events on the dot blank flip-flops IC32-2, 4, 5, and 8 (74LS74) 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.

5.3 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.

5.3.1 Time calibration

In verifying the timing diagrams related to horizontal sweep rates, the 4½ microsecond wide bars on IC1-14 (74161) 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 IC1-14 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.

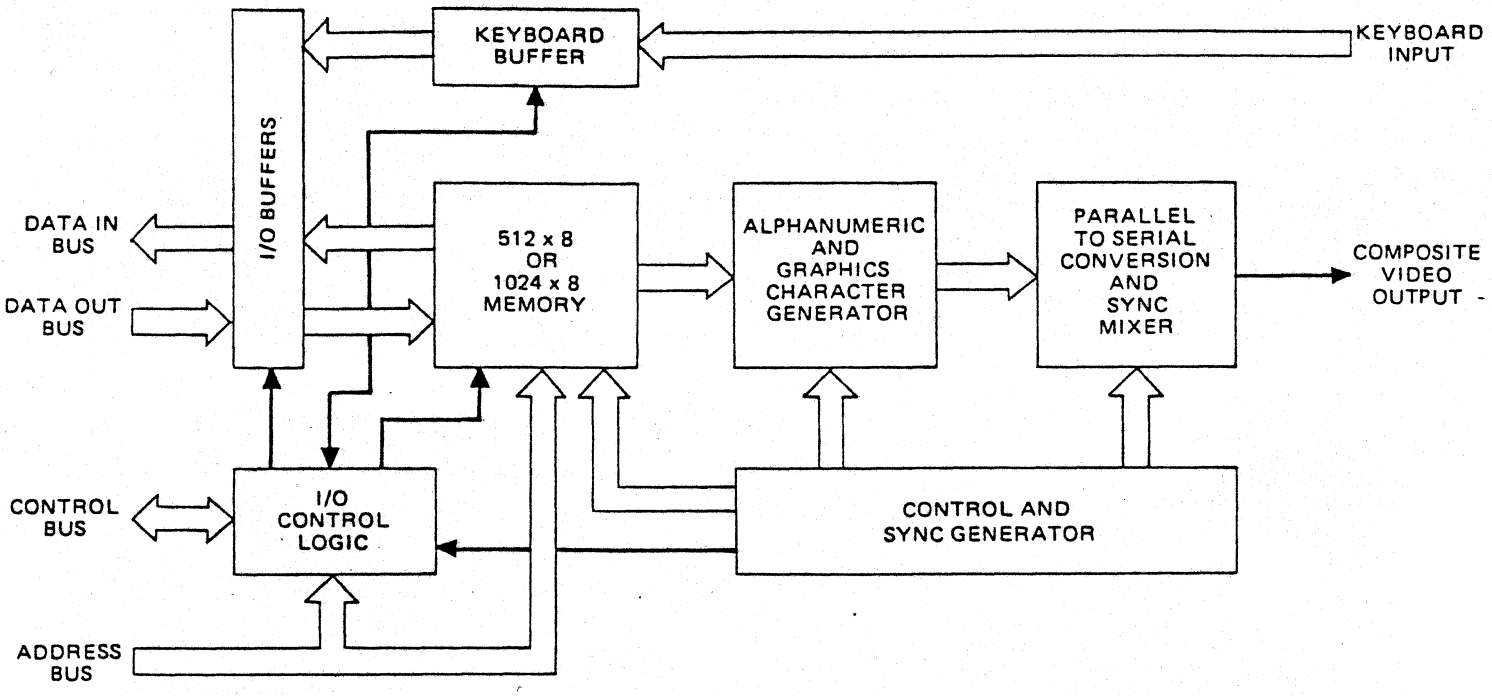
Calibration of the vertical dimension or vertical sweep time base is perhaps easiest by looking at IC15-3 (74393). The leading edges (measuring top to bottom) of the groups of white sweeps are 15 sweeps or 877 microseconds apart. A 16 line (240 sweep) visible raster is 14.04 milliseconds, and vertical 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).

6. VTI theory of operation and block diagrams

The principal functional blocks which form the video terminal interface are shown in figure B-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, interfacing with an input of 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.



A more detailed view of the board circuitry is shown in the schematic diagram at the end of this 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 S-100 data bus. We will be referring to the video terminal interface (VTI) data bus as the on-board bus, and the S-100 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.

6.2 Symbol generation

With a low on the OE (output enable) line from IC9 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 IC40 whenever there is coincidence (in IC30) of a dot pulse from the dot clock IC29 and an "end of character" (EOC) signal (tenth dot carry) from the "dot counter" IC13. In the absence of a one in the MSB (most significant bit) from the latch, MUX's (multiplexors) IC33 and IC36 pass the seven-dot conversion pattern of this display data from the character-generating ROM (read-only memory) IC37 to the least significant bits of the output shift register IC35. 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 IC38 and IC39. 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 IC15. The sweep counter is self-resetting after every fifteenth sweep, and this resetting action is accumulated in the line counter half of IC15.

In similar fashion, the dot counter IC13 is self-resetting every tenth dot, and its output is accumulated in the symbol counter IC16. The combination of line and symbol counter outputs determine the address of each individual character stored in the memory (IC's 20 through 28). 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.

6.3 Raster and 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 IC29, and blanking is maintained by a variable-duration one-shot IC34. 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 IC1 and again by thirteen or fourteen in IC2. A 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 IC2 so that it may again divide by 13 or 14. This binary 3 at the IC2 outputs will therefore last for one-thirteenth or fourteenth of the period between carries and is passed through IC3a 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 (IC15a) which is used both to address the character generation ROM and to signal the line counter ICI5b every fifteen sweeps that a new display line is being addressed.

When 16 line counts (16 X 15 = 240 sweeps) have accumulated in IC15b, the carry resulting from the transition from its binary 15 state to its binary zero state is inverted by IC5 to set the vertical blanking flip-flop IC4. In addition to blanking the screen,

IC4 also enables the 1 of 8 decoder IC12. After eight blanked sweeps have been counted by the sweep counter IC15, Pin 14 of IC12 will go low, producing a vertical sync pulse.

This vertical sync lasts the seven more lines until IC15a resets itself and advances the line counter. IC3 ANDs this vertical sync with the horizontal sync carry, so that the interruptions in the wide vertical sync pulse maintain horizontal sync.

Further subcounts fo the sweep and advances of the line counter accumulate in IC15 until IC12 decodes the 37th blanked sweep to trigger the pulse stretcher IC34. (Line counter = 2 and sweep counter - 7.) IC34 is a very short duration one-shot which terminates the vertical blanking (disabling IC12) and also resets the sweep and line counters for top of the page addressing. The subsequent termination of horizontal blanking has the character counter IC16 reset to prepare all addressing from the top left of page as described below.

6.4 Symbol and raster synchronization

Termination of the horizontal blanking one-shot IC34a re-enables the dot clock oscillator IC29a but does not unblank the screen. At this time, symbol count addresses are set to zero, but the data latch IC40 contains unrelated data sampled with some previous address. Similarly, the shift register IC35 contains old data. The screen has been darkened by the dot blank flip-flops of IC32 which have been held set by the horizontal blanking. The symbol counter IC16 MSB is presenting a zero to the D input of flip-flop IC32, 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 IC13 sends load signals gated through IC30 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 IC16 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 OEC 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 accumulated 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.

6.5 External bus and keyboard interfacing

The comparator IC6 compares the 6MSB's of the external address bus with the jumper pattern selected for display memory addressing.

In the switched condition, RAM address is determined by the ten LSB's on the external address bus instead of by the combination of the line and sumbol 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 IC41 will be sent to the CPU via the line drivers. The MEMR+ singnal, 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 IC7's to transfer the external data bus to the internal data bus and write it into the on-board RAM. In this way, CPU data can be written into display addresses, keyboard data can be input to the CPU. Keyboard data can be latched into IC41 in repsonse to "key pressed" strobes of jumper selected polarity. A jumper pattern to pin 4 of the external bus permits sending an interrupt request to the CPU when the latch IC41 is updated by a "key pressed" strobe.

7. Software

7.1 Video Typewriter:

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 or a symbol, 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.

This program when executed at address ØØØØ 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.

---

*This program assumes the user has a defined stack area. If you have no preassigned stack location, execute a LXI SP, ØFFFH.

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 address 1DØØH. 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.

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.

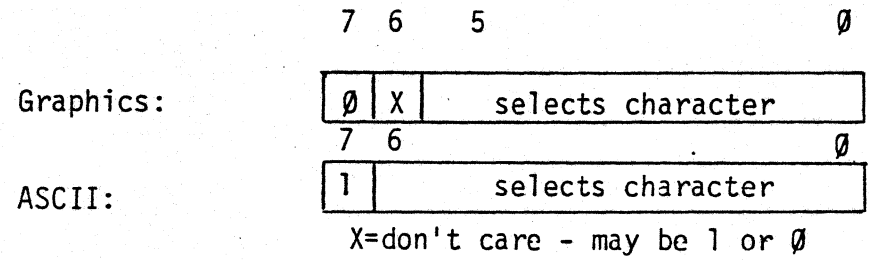
7.2 Graphics

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

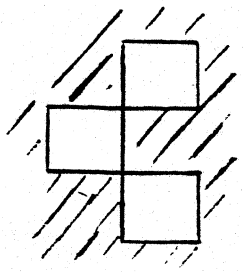
When a screen location is part of a graphics display, it is subdivided into six parts, thus:

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 Ø11Ø1Ø1ØB (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 hex 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 both programs at the addresses indicated. Execute the screen clearing routine at ØFØØ. If your system has a stack area already allocated, then you need not set the stack pointer. If the stack is not already initialized, set it with a LXI SP, ØFFFH. Then you are ready to load an initial generation (by using the hex-to-equal-graphic table in appendix D) memory locations in the middle of the screen (such as 8AlØH). When you are satisfied with your initial organism, execute the LIFE routine at address zero.

STORE AT 3800 RUN A 2000

Video Typewriter Routine STORE AT 3800 RUN AT 2000

Hexidecimal Address	Op Code	Mnemonic Instruction	Comments
0000		0100 SCRN-EQU 8800H F800	*VIDEO SCREEN ADDRESS
0000		0110 STR-EQU 1CFFH 3CFF	*STORAGE FOR SYMBOL UNDER CURSOI
0000		0120 STS-EQU 1CFEH 3CFE	*STORE OUTPUT MODE
0000		0130 CURS-EQU 1CFCH 3CFC	*STORE RELATIVE CURSOR LOCATION
0000		0140 SEND-EQU 8CH FC	*1ST BYTE OF SCREEN END
0000		0150 LINE-EQU 64	*LINE LENGTH
0000		0160 CS-EQU 0FFH	*CURSOR SYMBOL (RUB OUT)
0000		0170 LT-EQU 3FH	*LINE TERMINATION CHARACTER
0000		0180 KBD-EQU 88H F8	*KEYBOARD PORT ON VTI
0000		0190 ORG 0000 2000 7000	
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 7
0038	DB 88	0340 IN-IN KBD	*INTERRUPT DRIVEN KEYBOARD
003A	F6 80	0345 ORI 80H	
003C	F6 80	0350 ORI 80H	
003E	47	0360 MOV B,A	
003F	CD 00 1D	0370 CALL OUT	
0042	78	0380 MOV A,B	
0043	C9	0400 RET	
0044		0500 ORG 1D00H 3D00	
1D00	2A FC 1C	1000 OUT-LHLD CURS	
1D03	EB	1010 XCHG	*PUT RELATIVE CURSOR IN D
1D04	21 00 88	1020 LXI H,SCRN	*PUT SCREEN BLOCK ADDRESS IN H
1D07	19	1030 DAD D	*GET ABS CURSOR LOCATION
1D08	47	1040 MOV B,A	
1D09	3A FF 1C	1050 LDA STR	
1D0C	77	1060 MOV M,A	*PUT BACK CHAR UNDER CURSOR
1D0D	78	1070 MOV A,B	*CHECK*
1D0E	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 8C	1180 CPI 8CH	*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
1D36 FE 94      1240  CPI 94H         *CTL T FOR TEXT (X I/D)
1D38 CA B1 1D   1250  JZ RID
1D3B FE 86      1260  CPI 86H         *CTL F FOR FEED (SET ALF)
1D3D CA BC 1D   1270  JZ SALF
1D40 FE 8E      1271  CPI 8EH         *CTL N FOR NORMAL TTY (X ALF
1D42 CA C7 1D   1272  JZ RALF
1D45 FE 93      1280  CPI 93H         *CTL S FOR SCROLL (SET SCRL)
1D47 CA D2 1D   1290  JZ SSC
1D4A FE 90      1300  CPI 90H         *CTL P FOR PAGE (X SCRL)
1D4C CA DD 1D   1310  JZ RSC
1D4F FE 8A      1320  CPI 8AH         *LINE FEED
1D51 CA E8 1D   1330  JZ LF
1D54 FE 8D      1340  CPI 8DH         *CARRIAGE RETURN
1D56 CA 21 1E   1350  JZ CR
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 RO-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
```

```
1DA8 C2 A0 1D    2370  JNZ SWAP
1DAB 2B          2380  DCX H
1DAC 36 7F       2390  MVI M,7FH
1DAE 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
1DCA 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  ANA L
1E2E C2 28 1E    2960  JNZ SLOP
1E31 2B          2970  DCX H
1E32 3E C0       2980 BACK-MVI A,0C0H    *GO TO BEGINNING OF LINE
1E34 A3          2990  ANA E
1E35 5F          3000  MOV E,A
1E36 3A FE 1C    3020  LDA STS
1E39 17          3030  RAL
1E3A 17          3040  RAL
1E3B DA E8 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/O
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  ANA 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 C9          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      *INDEX BY SCREEN ADDRESS
1E7C 19          8070  DAD D
1E7D 7E          8080  MOV A,M         *STORE CHAR UNDER CURSOR
1E7E 32 FF 1C    8090  STA STR
1E81 36 FF       8100  MVI M,CS        *STUFF NEW CURSOR SYMBOL
1E83 C9          8110  RET
```

Life for the VTI

```
0000                    0100 VADD-EQU 8800H F800   *VIDEO BLOCK ADDRESS
0000                    0110 MADD-EQU 0300H 2300   *MASTER COPY ADDRESS
0000                    0120 SADD-EQU 0800H 2900   *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 2208   *TABLE (MASK & SCRATCH)
0000                    0170 TAD-EQU 02H           *1ST BYTE OF TADD
0000                    0175 CADD-EQU 0180H 2180   *COUNT ADDRESS (GENERATIONS)
0000 21 08 02           0180  LXI H,TADD          *SET UP MASK TABLE
0003 3E 20              0185  MVI A,20H           *FIRST MASK FOR TABLE
0005 0E 08              0190 MASK MVI C,08H       *GETS EIGHT SPOTS
0007 77                 0200 TABLE MOV M,A
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 BIT
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 A,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 A,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 A,M
0034 12                 0314  STAX D
0035 23                 0316  INX H
0036 13                 0318  INX D
0037 7C                 0320  MOV A,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 88           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 A,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,A             *2ND BYTE FEEDS MASK TABLE
008A 1A          1260  LDAX D              *LOAD MASK FOR BIT
008B A6          1270  ANA 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 A,07H ?         *ADDRESS TO STORE NEIGHBOR
0092 A5          1310  ANA 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
```

Go To Pg. 64

```
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,VADD         *SWAP SLAVE TO SCREEN
00D3 C3 20 00    1830  JMP LOOP           *ON EACH SUCCESSIVE LOOP.
00D6 C4 65       1840 INSI 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
```

Go To pg 63

```
00EA C8 4C      1940  DW 4CC8H          *IN 3X3 MATRIX OF
00EC AC CC      1950  DW 0CCACH         *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
```

Go To Appendix A

APPENDIX A ASCII Character set

b7 b6 b5 / Bits b4 b3 b2 b1 / ROW / COLUMN	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
	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	[	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

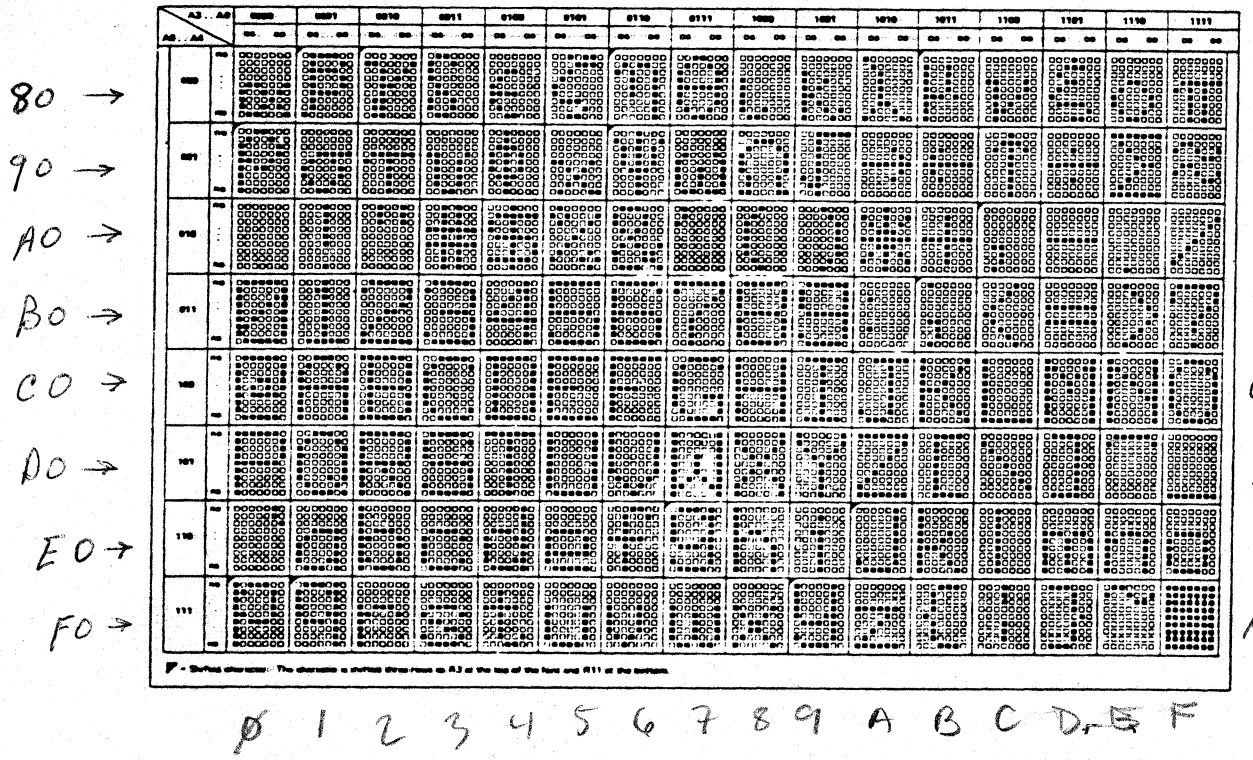
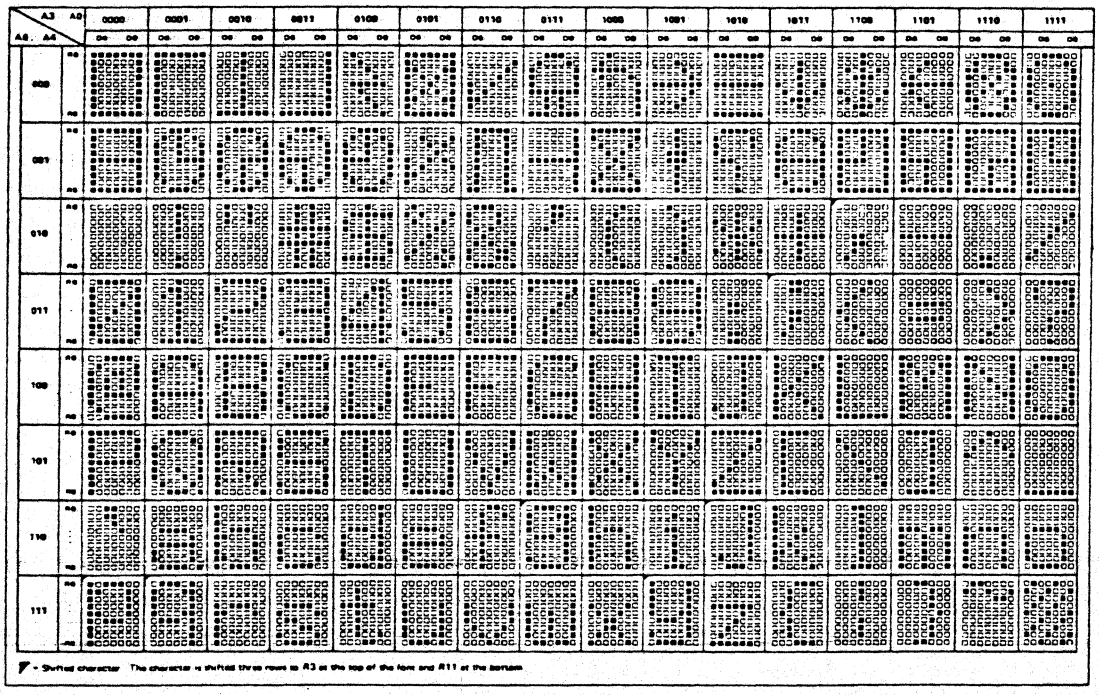


FIGURE 18 – MCM6574 PATTERN



APPENDIX B Con't

FIGURE 19 – MCM6575 PATTERN

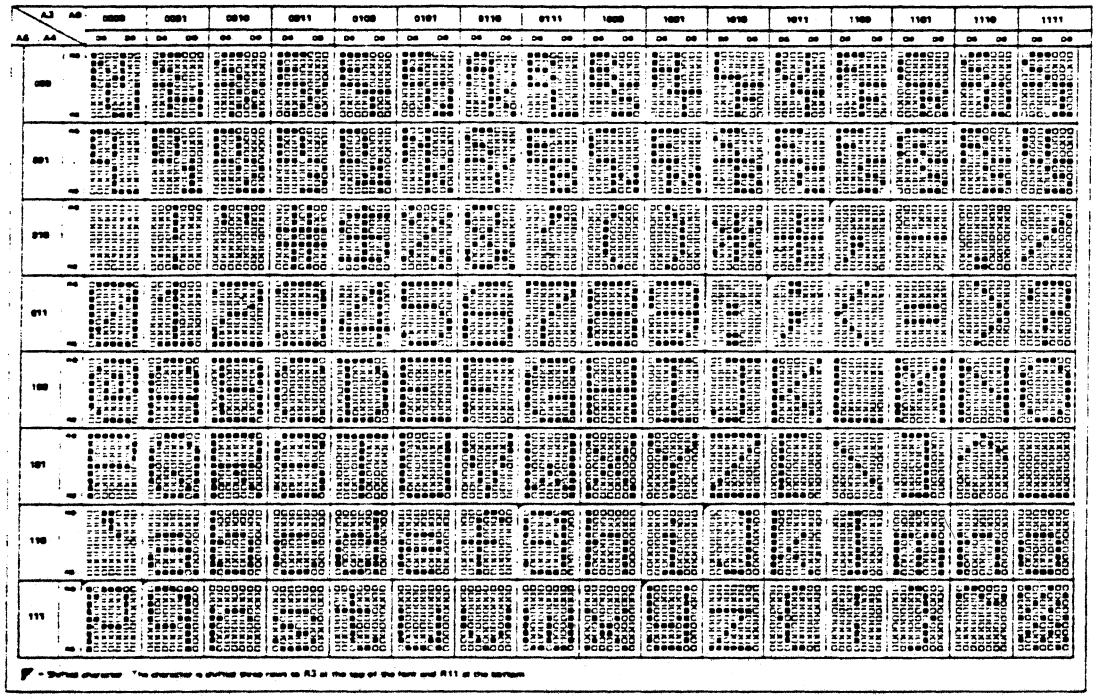
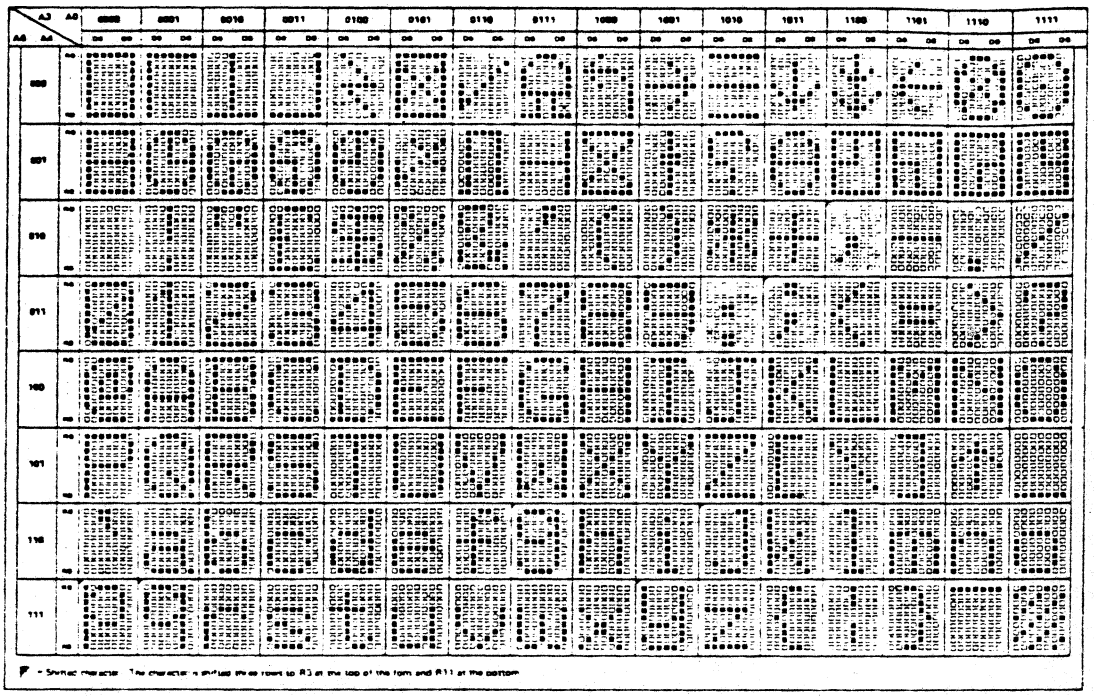


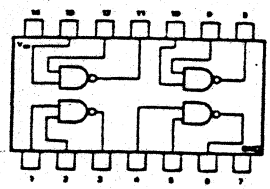
FIGURE 20 – MCM6576 PATTERN



APPENDIX C Chip pinouts

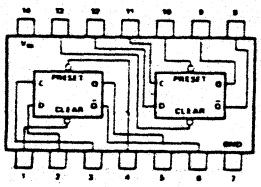
A,F PACKAGE

7400

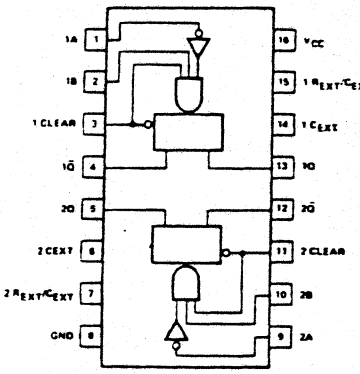


A,F PACKAGE

7474

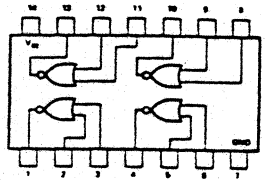


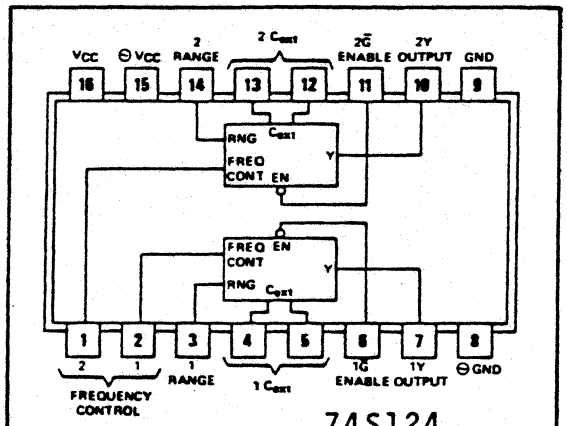
74123



A,F PACKAGE

7402



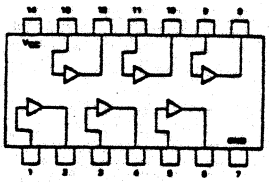


74S124

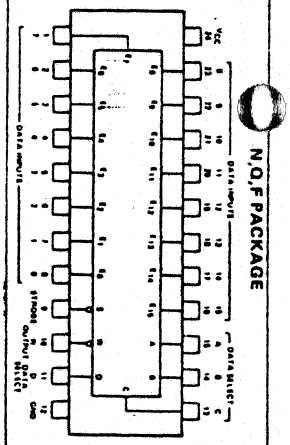
positive logic: While the enable input is low, the output is enabled. While the enable input is high, the output is high.

A,F PACKAGE

7407



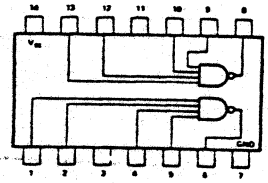
74150



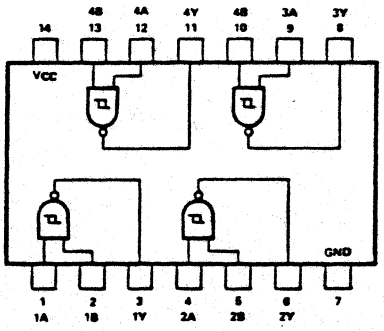
N,J,F PACKAGE

A,F PACKAGE

7420



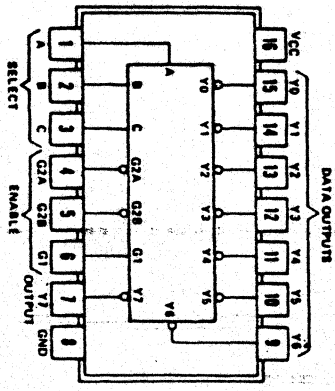
S54132, N74132



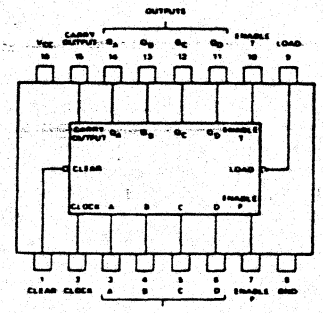
POSITIVE LOGIC: $Y=\overline{AB}$

74LS138

positive logic: see function table

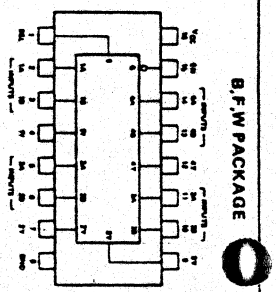


B,F,W PACKAGE



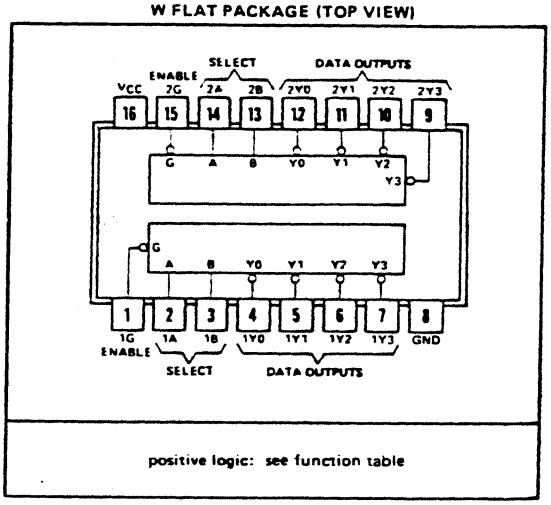
74161

74157



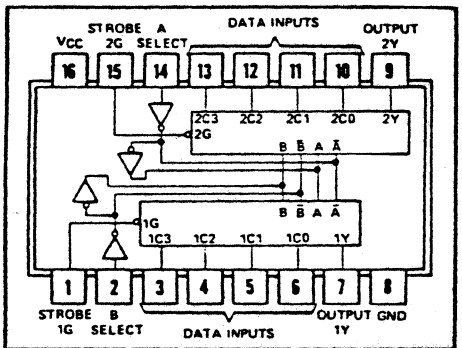
B,F,W PACKAGE

'LS139, 'S139
J OR N DUAL-IN-LINE OR
W FLAT PACKAGE (TOP VIEW)



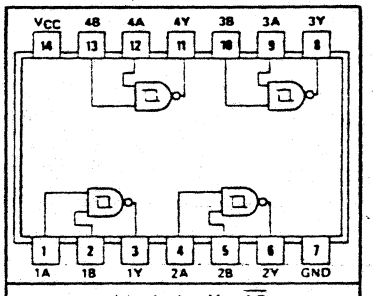
positive logic: see function table

'153, 'LS153, 'S153 . . . J, N, OR W PACKAGE
'L153 . . . J OR N PACKAGE
(TOP VIEW)



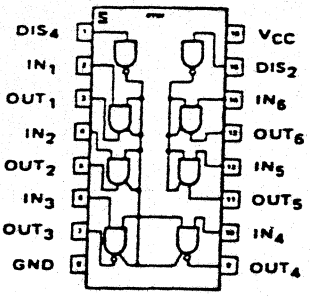
positive logic: see function table

SN54LS132 . . . J OR W PACKAGE
SN74LS132 . . . J OR N PACKAGE
(TOP VIEW)

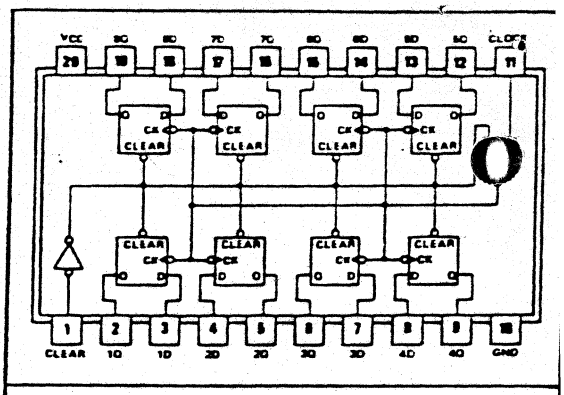


positive logic: $Y = \overline{AB}$

74367
or 8097

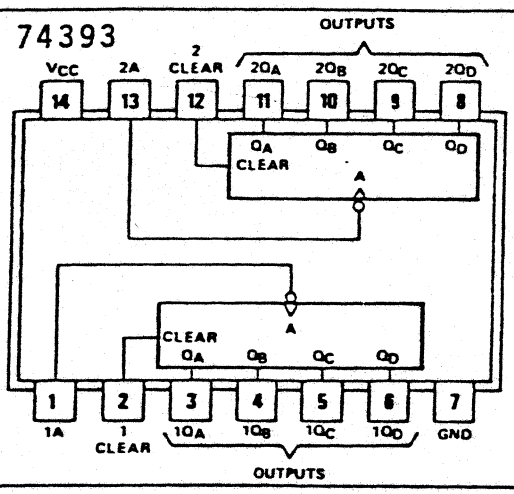


74273



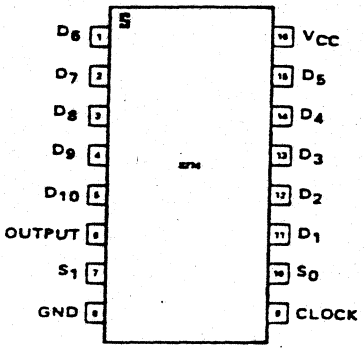
positive logic: see function table

74393

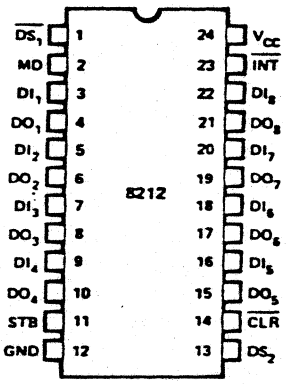


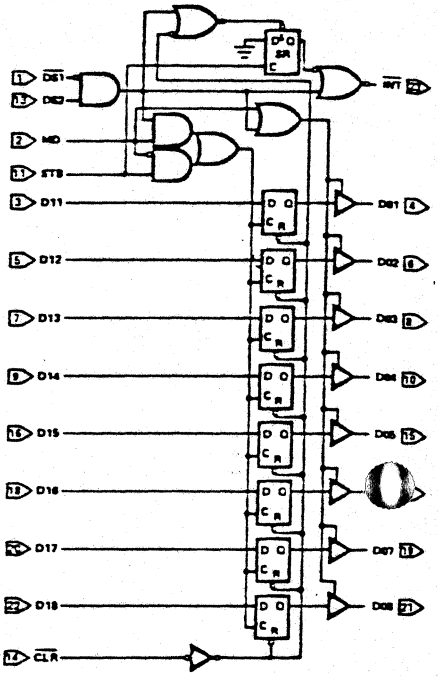
positive logic: High input to clear resets all four outputs low

N8274B

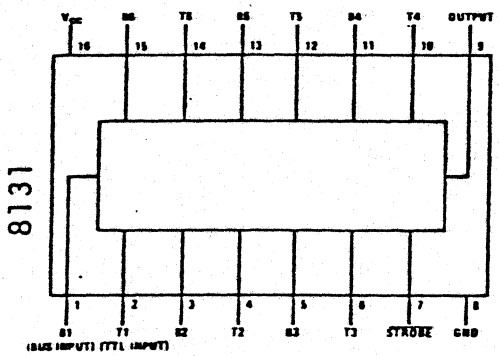


8212 or
74S412



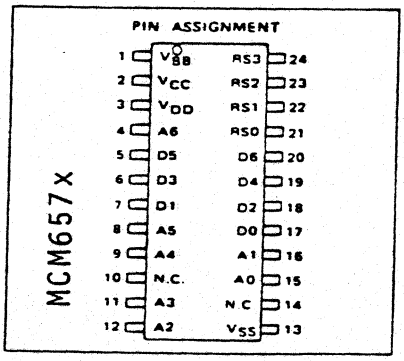


8131

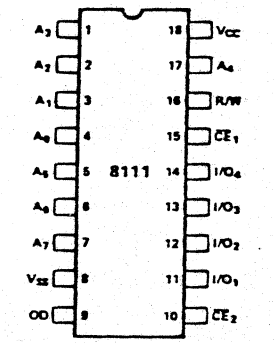


7131(J), (W); 8131(J), (N), (W);
7136(J), (W); 8136(J), (N), (W)

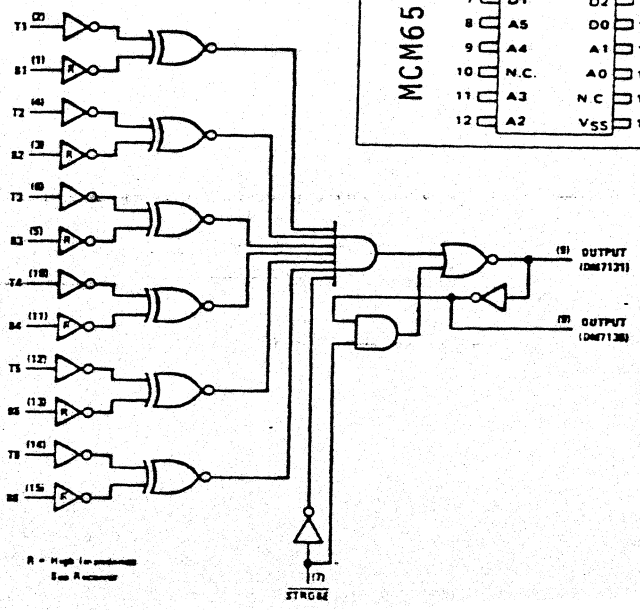
MCM657x

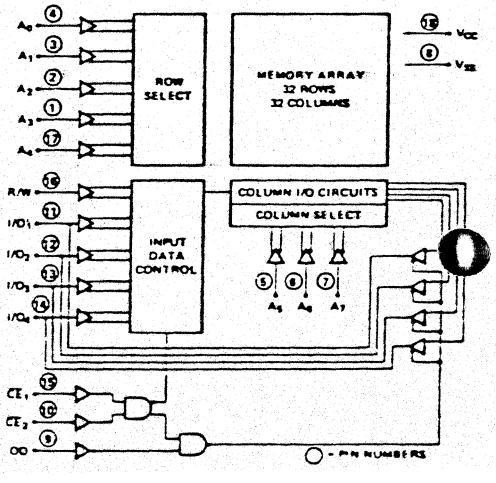


2111



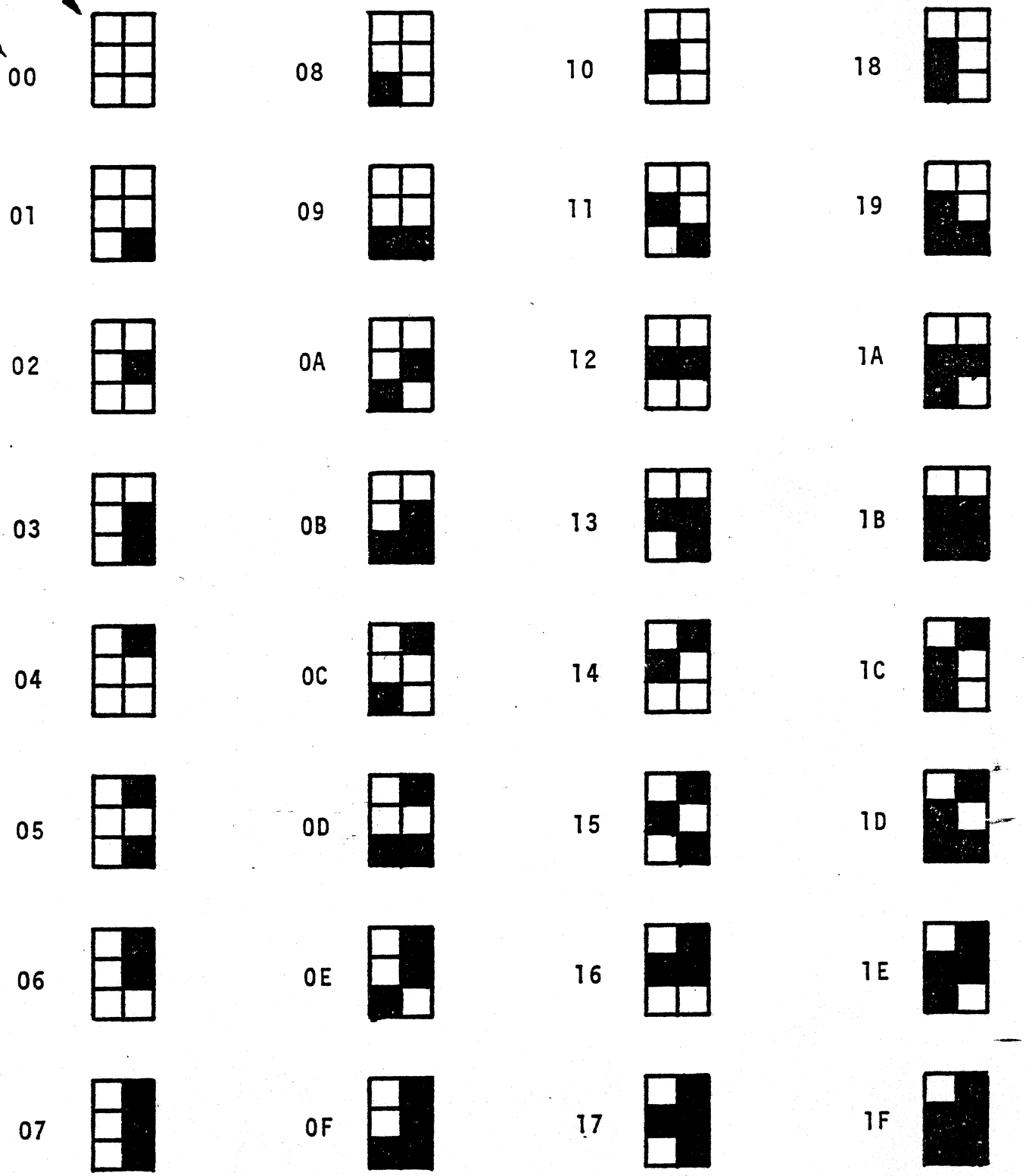
Logic Diagram



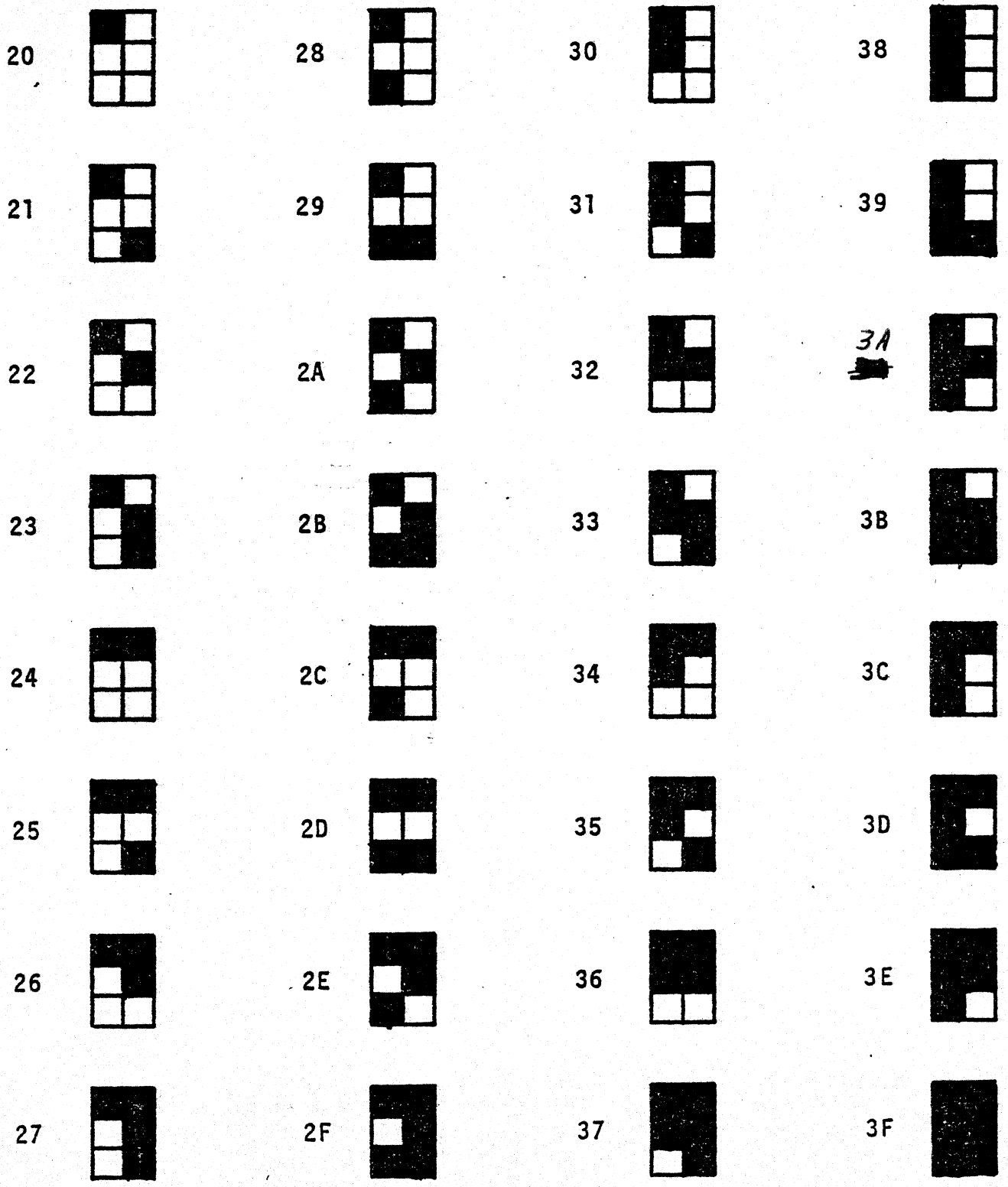


Appendix D Graphics character set

Hex Graphic (white bright, black dark)

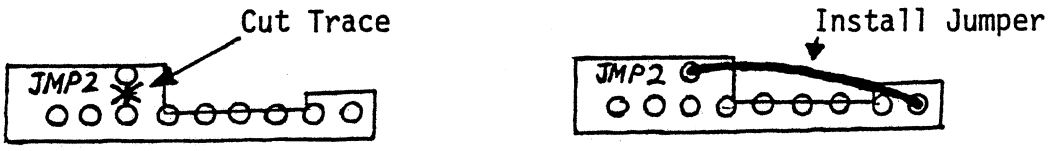


Appendix D Con't



1. Page 17, the JMP 1 figures for "Other S-100" and "POLY 88" are reversed Use the left figure for the POLY 88 and the right for the other S-100. Note the S-100 bus edge of the card is to the top of all three JMP 1 figures.

2. JMP 2 is not discussed in the manual. The wiring depends on the intended use. Most non-POLY 88 applications do not have vectored interrupt. If you do not have vectored interrupt, but wish to use the VTI keyboard port with interrupts, cut the PC jumper and install a jumper as shown below.

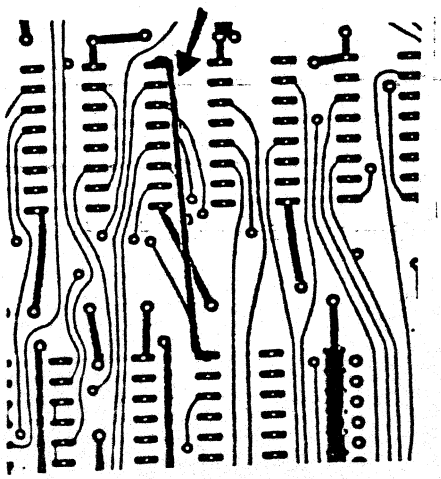


3. Page 21, the two sentences just before section 4.5 should read:

If your keyboard is type 1 or 3, the jumper is already configured correctly.

If it is a type 2 or 4, cut the minus trace from the center pad of JMP 7 and jumper from center pad to + labeled pad.

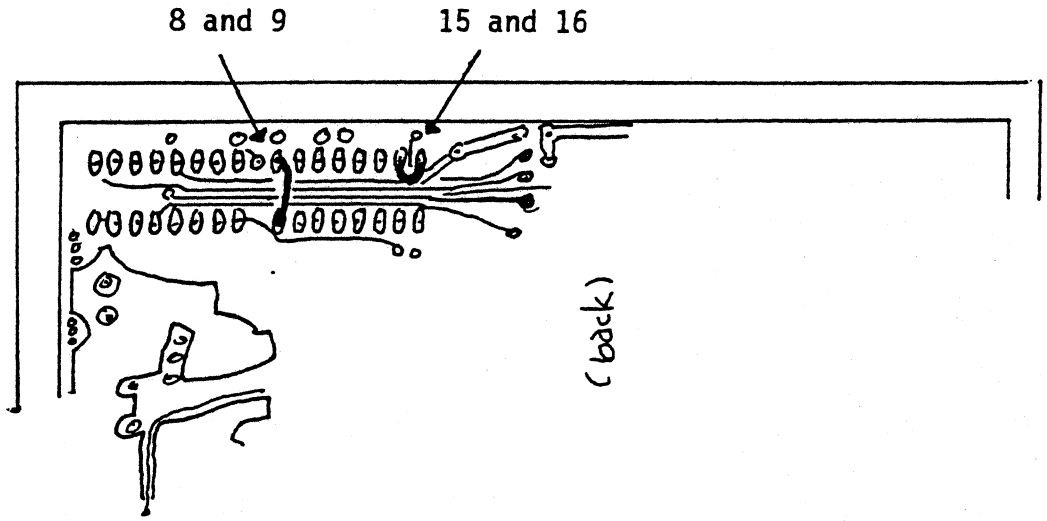
4. There is an artwork error on the video board which requires modification. Adjacent to pin 16 of IC 19 is a trace which drops through a plated through hole from the front. Cut the trace just where it attaches to the feed-through hole on the front of the board. Attach a jumper from pin 1 on IC 19 to pin 9 on IC 6 as shown below:

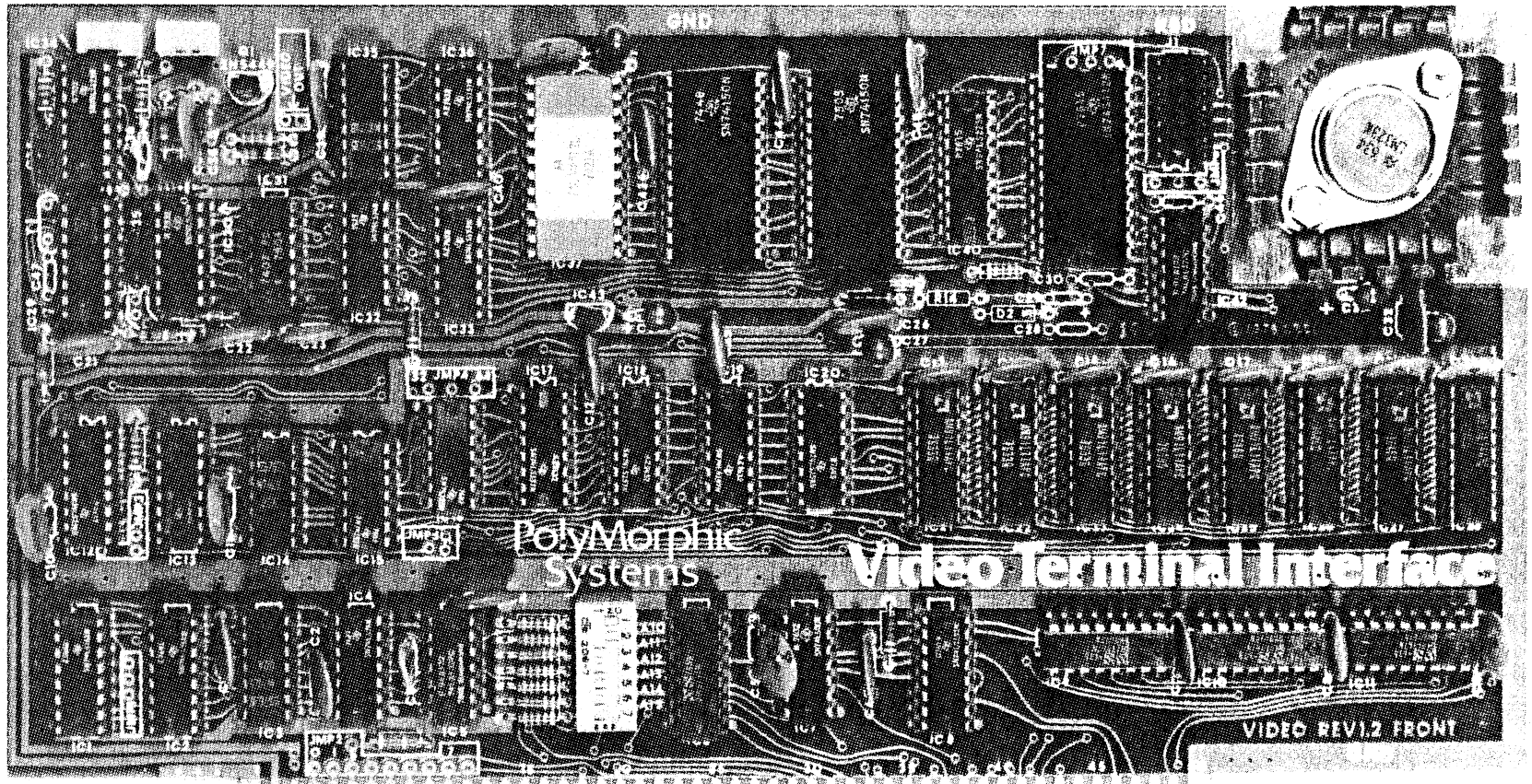


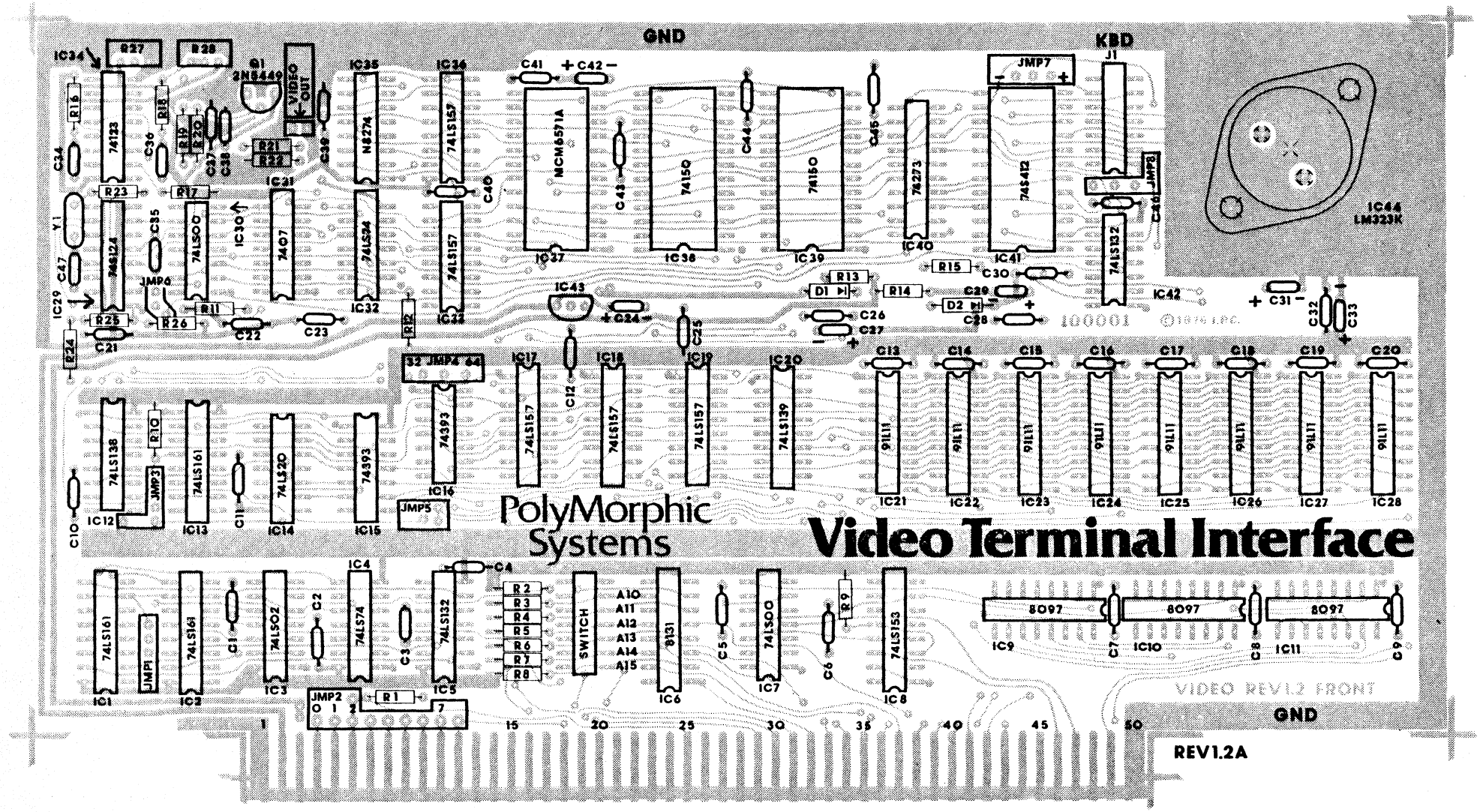
(view shows back side of video board)

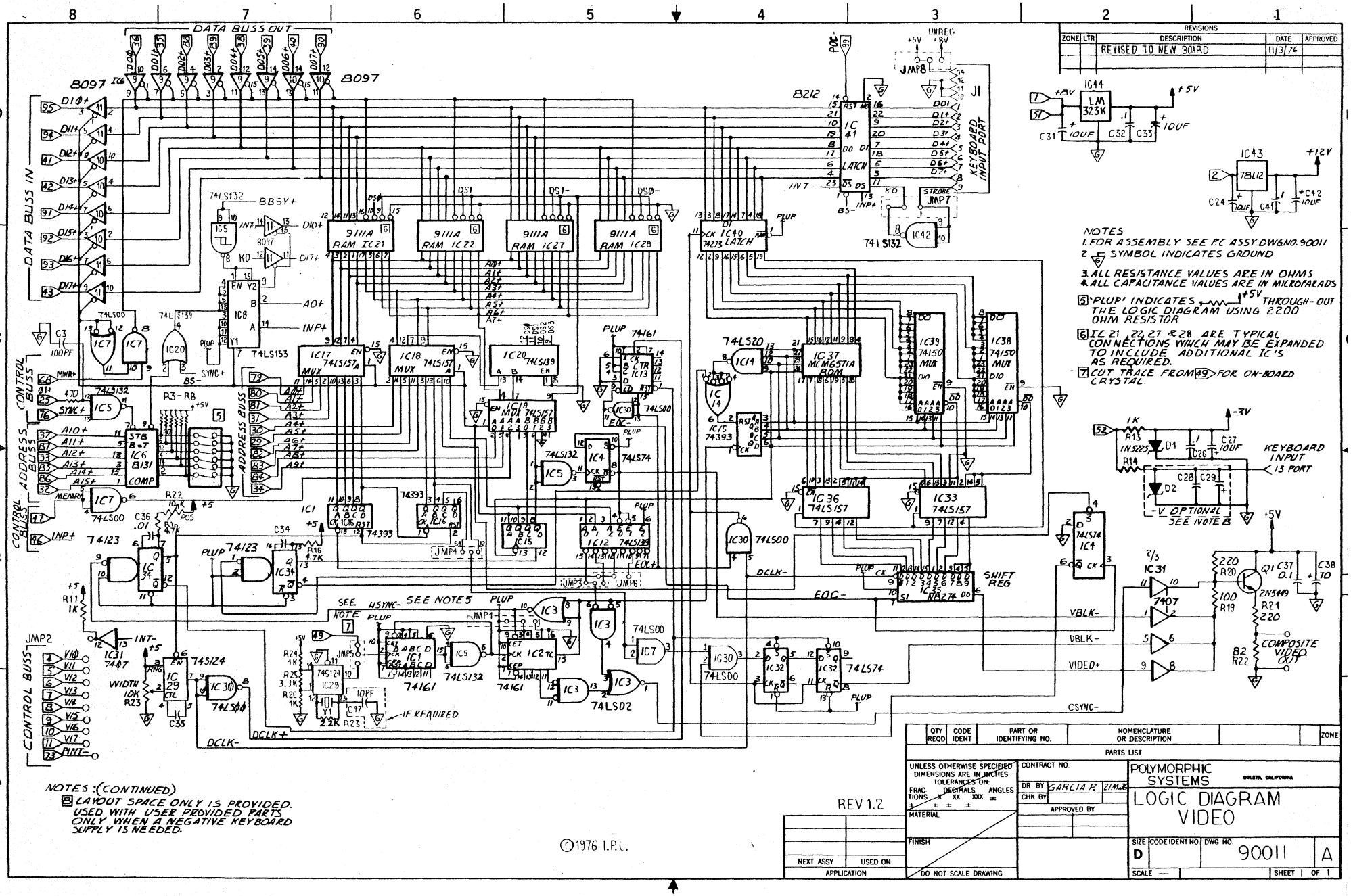
5. There is an artwork error on the video board which requires a modification.

On the back side of the board, jumper pins 8 and 9 of IC 29 together, with #24 wire and insulating tubing. Similarly, jumper pins 15 and 16 of IC 29 together.




PolyMorphic
Systems
Video Terminal Interface
VIDEO REV1.2 FRONT


GND
KBD
J1
JMP7
IC44
LM323K
PolyMorphic
Systems
Video Terminal Interface
VIDEO REV1.2 FRONT
GND
REV1.2A



REVISIONS				
ZONE	LTR	DESCRIPTION	DATE	APPROVED
		REVISED TO NEW BOARD	11/3/76	

NOTES
1. FOR ASSEMBLY SEE PC ASSY DWG NO. 90011
2. ⏚ SYMBOL INDICATES GROUND
3. ALL RESISTANCE VALUES ARE IN OHMS
4. ALL CAPACITANCE VALUES ARE IN MICROFARADS
5. 'PLUP' INDICATES +5V THROUGH-OUT THE LOGIC DIAGRAM USING 2200 OHM RESISTOR
6. IC 21, 22, 27 & 28 ARE TYPICAL CONNECTIONS WHICH MAY BE EXPANDED TO INCLUDE ADDITIONAL IC'S AS REQUIRED.
7. CUT TRACE FROM 49 FOR ON-BOARD CRYSTAL.

NOTES: (CONTINUED)
8. LAYOUT SPACE ONLY IS PROVIDED. USED WITH USER PROVIDED PARTS ONLY WHEN A NEGATIVE KEYBOARD SUPPLY IS NEEDED.

REV 1.2

QTY REQD	CODE IDENT	PART OR IDENTIFYING NO.	NOMENCLATURE OR DESCRIPTION	ZONE

PARTS LIST

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES. TOLERANCES ON: FRACTIONS ± DECIMALS .X ± .XX ± .XXX ± ANGLES ±

MATERIAL

FINISH

DO NOT SCALE DRAWING

CONTRACT NO.

DR BY GARCIA P. 2/MAR

CHK BY

APPROVED BY

NEXT ASSY | USED ON

APPLICATION

POLYMORPHIC SYSTEMS GOLETA, CALIFORNIA

LOGIC DIAGRAM VIDEO

SIZE D | CODE IDENT NO. | DWG NO. 90011 | A

SCALE — | SHEET 1 OF 1