THE VDB™ MANUAL

V D B

Video Display Board
Preliminary User's Manual

(Manual Revision 0)
November 1977

Technical Design Labs, Inc.
Research Park, Bldg. H
1101 State Road
Princeton, New Jersey 08540
609-921-0321

WARNING

DANGER

Connecting the VDB to a hot chassis television set, (i.e. 110 volts) will cause destruction of the ground traces on the PC boards, damage to IC's, and may even damage other PC boards in your computer.

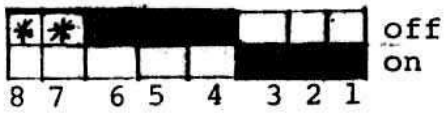
Please be advised that this action will void the warranty and in this case TDL will not be responsible for damage to other equipment.

How to Set the VDB Sense Switches

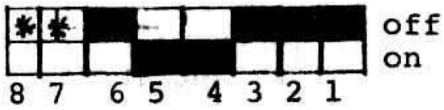
Looking at the VDB switches as they appear (upside down).

1. For the RAM Driver Programs-Ports
   EØ-E3

*=Don't Care
■=IN



2. For the EPROM Driver Program-Ports
   9C-9F



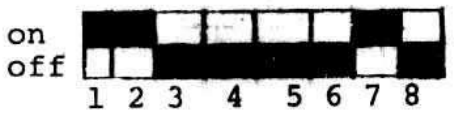
- Note that the Driver Programs require a jumper between points A and B (front board).

How to Set SMB + SMB2 Sense Switches for VDB as Console

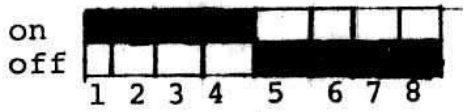
1. For SMB see VDB boot flier.

2. For SMB2 (with EPROM) switches upright.

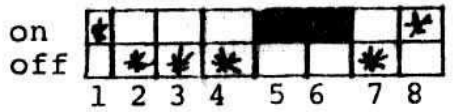
SW 1



SW 2



SW 3



- Please note as stated in the SMB2 manual that at SW 3-switches # 7&8 should not be on at the same time.
- also note that the positioning on the "don't care" (=IN) are as recommended for reading our cassette tapes at 1200 Baud.

- The position of SW 3#3 (Cassette Reading Polarity) may have to be changed depending on your tape recorder.

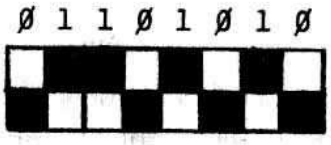
How to use the VDB boot cassette

These are the steps required to use the self-booting feature of the VDB driver supplied on TDL cassette.

1. Set video port to 9600 BAUD RATE. *Note: Must be 2K of RAM in SMB.

2. Set I/O configuration switches to:

   console = BATCH MODE
   reader = CASSETTE
   punch = (CASSETTE)
   list = VIDEO



■ = IN

3. Hook SMB cassette input to speaker output of cassette player.

4. Adjust volume to slightly piercing (unplug speaker jack for some units).

5. Rewind cassette to beginning.

6. Turn on Xitan computer.

7. Press reset button. Video monitor on VDB should show miscellaneous data.

8. Start cassette. Wait ≈ 2 minutes. If load was successful, go to step #10.

9. Switch "INVERT" switch on cassette (see SMB manual) & go to step #5.

10. VDB will go clear, and "ZAPPLE V1.0" will sign on.

11. The console input will be the VDB parallel input port.

12. Set "CONSOLE" configuration to "USER" mode on I/O configuration switches. (Extreme right switch position "IN" on top).

Table of Contents

Video Display Board (VDB)

A. Circuit Features

80 characters per line
25 lines per screen
Upper/lower case with descenders
64 graphic characters
Independent memory (not part of the processor's address space)
2 full screens of display memory
Software controlled character inversion
Software controlled character blink
Software controlled display inversion
Software controlled cursor
Software addressable cursor
Software readable cursor
Software readable display memory
Software controlled display inhibition
Works with modified TV set or monitor
On-board 8-bit parallel port for keyboard and status lights
Software drivers for 8080 or Z80 and RAM or ROM
S-100 bus compatible
2 boards (only one S100 connector needed)

B. Specifications

1. Introduction:

The VDB interfaces to an S-100 bus and provides alphanumeric and graphic information in the form of composite video signal for driving an external CRT monitor. It also provides a parallel keyboard interface for input purpose and a set of eight latched output lines for driving external devices such as lights and buzzers.

2. Physical Size:

The VDB occupies one S-100 bus socket and takes up two slots of space.

3. Buffer Memory and Display Format:

The VDB contains two pages of onboard buffer memory. Each page is organized as 25 data rows of 80 symbols to each data row. One page is displayed at a time.

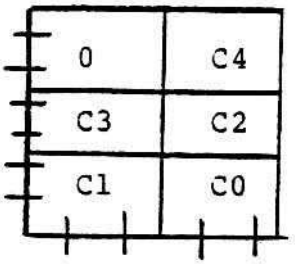
4. Display Symbols:

Each displayed symbol is stored as an 8-bit code C7 C6 C5 C4 C3 C2 C1 C0 in the buffer memory. The 96 non-control characters of the ASCII set are displayed as alphanumeric symbols and the 32 control character codes are used for the graphic symbols. Since ASCII code requires only the low order seven bits, the MSB C7 is used according to the following rules to modify the displayed symbols:

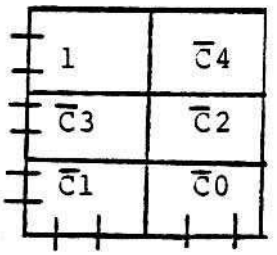
a. For the 32 control symbols, C7=1 will invert the symbol display, giving a total of 64 different graphic symbols.

b. For the 96 non-control symbols, C7=1 will invert the symbol if the mode register invert bit is enabled, and blink the symbol if the mode register blink bit is enabled, and invert and blink if both are enabled.

Each symbol is allotted a 6x9 space on the display screen. For the 96 alphanumeric symbols the top row and the left column are not used so as to provide inter-character spacing. The 32 graphic symbols use the entire 6x9 space so graphs can be formed with no inter-symbol gaps. The graphic capability may be stated as having 160 horizontal elements and 75 vertical elements. The mapping of the bits for the graphic symbols are shown below: Note: C6 C5= 00.



For C7=0

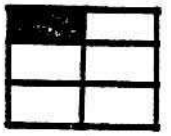


For C7=1

For example:

The character 00011111 is displayed as 

The character 10000000 is displayed as 

The character 10011111 is displayed as 

5. Cursor:

A cursor is provided to indicate on the display screen the physical location where the next symbol can be written into or read from. The cursor is in the form of a rectangular block, alternating with the symbol at the given location with a rate of 3.75 Hz.

6. Mode Register:

The VDB has a six-bit mode register, M= M5 M4 M3 M2 M1 M0 for modifying the display characteristics.

M0. page select:	0 selects page 0, 1 selects page 1.
M1. blink enable:	0 disables blink, 1 enables blink.
M2. invert symbol:	0 gives normal symbols, 1 gives inverted symbols.
M3. invert display:	0 gives black background. 1 gives white background.
M4. innibit cursor:	0 enables cursor display, 1 inhibits cursor display.
M5. inhibit display:	0 enables display, 1 inhibits display.

Mode Register bits M2 and M3 interacts with C7 in the manner shown in the following chart:

		M3=0 Black page background.	M3=1 White page background.
96 Alpha numeric symbols	M2•C7=0	White symbols,black symbol space background.	Black symbols, white symbol space background.
	M2•C7=1	Black symbols, white symbol space background.	White symbols, black symbol space background.
32 graphic symbols	C7=0	White symbols, black symbol space background.	Black symbols, white symbol space background.
	C7=1	Black symbols, white symbol space background.	White symbols, black symbol space background.

NOTE: M2 has no effect on the graphic symbols.

Blink, when enabled, causes the symbol and the symbol space background to blink at a rate of approximately 4 Hz.

7. Communication with S-100 Bus:

The VDB uses 4 I/O ports with addresses A7 A6 A5 A4 A3 A2 X X, where X X runs 00 to 11. The a's are selectable with on-board switches.

XX = 00 is the display control port.

XX = 01 is the display data port.

XX = 10 is the keyboard control port(IN)
external port(OUT).

XX = 11 is the keyboard data port.

With the exception of XX = 11 which is an input port only, all others are bi-directional.

a. Display Data Port: An output instruction to the data port places the data into the buffer memory and steps the cursor to the next sequential position. An input instruction reads the data in the buffer memory pointed by the cursor and then steps the cursor. There is no latency in the read operation. Sequential position is defined as follows:

Position (m,n)+1 is followed by:

[For m=0 thru 78]	(m+1,n)
[For m=79 and n=0 thru 23]	(0,n+1)
[For m=79 and n=24]	(0,0)

b. Display Control Port: The display control port is used to send to VDB the cursor, mode register data and to set up the subsequent reading of the cursor and mode register data. As an output port, the data bits are commands to VDB and have the following interpretations:

(i)	0xxxxxxx:	Load the X-cursor register with xxxxxxx.
(ii)	10mmmmmm:	Load the mode register with mmmmmm.
(iii)	110yyyyy:	Load the Y-cursor register with yyyyy.
(iv)	111---00:	enable X-cursor read
(v)	111---01:	enable Y-cursor read
(vi)	111---10:	enable mode register read
(vii)	111---11:	reset the VDB so that mode register, X-cursor and Y-cursor are all reset to o's. For iv, v, vi, vii (-) means either 0 or 1. (Don't care).

I/O instructions to the display takes approximately 2 microseconds to complete for VDB manufactured to operate in 60 Hz power networks, and approximately 2.5 microseconds in 50 Hz power networks.

c. Keyboard Data Port: Through this port the keyboard input data may be read. Upon reading, the status bit is set to 1.

d. Keyboard Control Port: As an input port, the MSB gives the status of the keyboard. When it is 0, it signifies that the keyboard has data to be read.

As an output port the eight bits are latched and brought to a connector for external use.

C. Principles of Operations

The VDB (Video Display Board) employs a 4000-character random access memory to store the display, which is organized into two pages of text, each consisting of 25 lines of 80 characters per line. One page is displayed at a time.

Each display symbol occupies a 6 X 9 matrix space. For the displayable ASCII characters, in general, a 5 X 7 subspace is used, with the leftmost column and the top and bottom rows unused in order to provide inter-character and line-to-line spacings. The graphic symbols employ the entire 6 X 9 matrix space, thus permitting continuous drawing capability within the graphic resolution limits of VDB.

The display uses a television raster format. The first visible video scan line contains the top slice of each of the 80 characters of the first text line. The second visible video scan line contains the second slice of the 80 characters of the first text line, etc. In this manner, each consecutive nine video scan lines provide all the slices necessary for each text line and the space separating consecutive text lines. After the last video scan line associated with the last text line has been traced, vertical retrace takes place, after which the display cycle repeats.

To generate each video line with the proper intensification, the 80 locations in the memory corresponding to the given text line must be accessed sequentially. The memory outputs, which define the character code, are combined with the video slice code and are used to address the character generator ROM (Read-Only-Memory). Output of each such access provides 6 bits of data which represent the intensifying information.

NOTE: (All *'ed items are on card A and non-*'ed items are on card B).

1. INPUT-OUTPUT INTERFACE:

The data output bus DO0 through DO7 are buffered by U7* and U8* to produce D0 through D7. Address lines A2 through A7 are compared with the DIP switch U20* settings by U18* and U19*. Signals A0 and A1 are further used to decode the address for display data port, display control port, keyboard data port and keyboard status port. Bus signal DBIN, SINP, PWR- and SOUT are used by U21*, through U25* to provide the read/write control signals for these ports. For example, the keyboard read control KBRD- enables U1* and U2* to present data from the keyboard data storage register U5* and U6* to the data-in bus DI0 through DI7. The keyboard status read control KBSRD- enables the keyboard status flip-flop U29* onto DI7 line. The

keyboard write control K3W- clocks the buffered output data D0 through D7 into the set of flip-flops of U11* and U12* for possible external use. The display data read control DATRD- enables the display data stored in U3* and U4* to be connected to the data input bus. The display data write control DATWR- clocks the buffered output data D0 throught D7 into U9* and U10* for entering into the buffer memory U30* through U37* at the proper time. Display control port read/write control signals CTLRD- and CTLWR- are used on card B for reading or writing into the cursor and mode registers.

2. DISPLAY LOGIC:

The display logic can be broken down into four sections. Namely, Raster Timing Section, Cursor Section, Memory Section and Video Signal Generator Section.

a. Raster Timing Section:

The crystal oscillator U17* provides the master clock BC at 11.36916 MHz. This is counted down by a factor of six by U20. The last count is decoded as B5 which is used with BC to generate the memory cycle clock CC. The RAM operates continuously at this rate and will be explained in more detail later. Memory cycle clock CC is counted down by a factor of 121 by U10 and U12 which form the Video X-counter. A complete cycle of the video X-counter defines the time duration given to each horizontal video scan line at the TV monitor. Of the 121 memory cycle intervals, 80 are used for display and the remaining ones are for horizontal retrace and overscan consideration. The Video X-counter is further counted down by a factor of 9 by U19 which is the Video Scan Counter. Each line of displayed text is allocated nine such consecutive video scan lines. RC occurs once at the end of each cycle of the Video Scan Counter. RC is further counted down by a factor of 29 by 1/2 of U7 and U8 which form the Video Y-counter. Thus one cycle of the Video Y-counter has 261 video scan lines and forms one frame of the display. Of the 261 video scan lines, 225 are used to display the 25 lines of text, and the remaining are used for vertical retrace and overscan consideration. The frame rate is identical to the AC line frequency to give flicker-free and steady image on the monitor.

The Video X-counter signals X1 through X64 and the Video Y-counter signals Y1 through Y16 define the column and row co-ordinate position of the displayed characters and are used in display buffer memory addressing. This will be described in Section c. HSYN-, generated by the X-signals, is the horizontal sync signal. U11 is used to provide adjustable horizontal position of the display image. Other signals, such as VSYN-, vertical sync, HBLK-,

horizonal blanking, VBLK-, vertical blanking and CSYNC, composite sync are derived from the X and Y signals also. Video signal, VIDEO- is mixed with CSYN and mode register bit M5 in U24 to generate the composite video output.

b. Cursor Section:

The cursor register is divided into the X-cursor U44 and U51 and the Y-cursor U42 and U49. When a display control port write instruction is given, CTLWR- is produced. If the most significant bit of output data, namely D7, is 0 then KWX-, cursor write-X, is generated. KWX- provides the clock and load enable signals to load D0 through D6 into the X-cursor register. If the most significant three bits of the output data, D7, D6 and D5 are 110, KWY-, cursor write-Y is generated, which loads D0 through D4 into the Y-cursor register in a similar manner.

The cursor register is used to define the location from which the data can be next read or into which a new character can be written. After such read or write the cursor is stepped to the next sequential position. Thus the cursor register is also a counter with the X-cursor register counts from 0 through 79 and the Y-cursor counter counts from 0 through 24. The Y-cursor counter is stepped when X-cursor steps from 79 to 0.

The cursor register also defined the position at which the cursor blinking should appear displayed on the monitor. U2, U4 and U6 form a comparator to compare the static cursor register outputs with the constantly changing video counter outputs to give the signal CURS. U25 and Y16 down by a factor of 16 to give the blink rate of 3.75 Hz. Signals CURS, BLINK and mode register output M4- are combined to produce signal A- which controls one input of U28 to satisfy the blink requirement.

To read the cursor registers (and the mode register), the display control port must be primed with an enable-read instruction. This is done with CTLWR- with the three high order data output bits, D7, D6 and D5 at 111, D0. D1 defines whether the subsequent data-in is for read X-cursor, Y-cursor or the mode register. U33 is the decoder and U34 is the command storage. The subsequent display data control read CTLRD- provides the read signals RDKX-, RDKY- and RDMR- which are tri-state control signals to connect 2/3 of U43 and U50 (X) or U48 and 1/3 of U50 (Y) or 1/3 of U43 and U47 (Mode) to the input data bus.

When it is desired to read the screen data, as described before, DATRD- is generated to tri-state

U3* and U4* onto the data input bus. When it is desired to write data into display, DATWR- is generated, which clocks buffered output data D0 through D7 into U9* and U10*. One half of U13* provides a one memory cycle synchronized version of DATWR- to enable the memory, U30* through U37*, read-write signal for write. DATRD- or DATWR- produces STEP- signal which causes the cursor register to step to the next sequential position. The synchronized DATWR-, DATRD- or a load cursor operation which produces KWXY- causes the other half of U13* to produce a synchronized signal on pin 8. This signal is used to produce RAM DATA STROBE signal at U15* pin 8 for loading U3* and U4* from the memory output. This arrangement lets U3* and U4* always containing the data as pointed to by the cursor register, whether it is immediately after a cursor load operation or after the cursor has been stepped following a data read or data write operation. U13* outputs on pin 6 and pin 8 are orred to give MAS, memory address select, to select the cursor registers instead of the video counters for addressing the memory.

c. Memory Section:

Memory IC's U30* through U37* are each a 4096-bit dynamic RAM.

The display buffer memory is normally addressed by the video counter outputs for display purpose. Only when it is necessary to perform a computer directed data read or write operation, the normal addressing sequence is suspended. MAS performs the selection with U1, U3 and U5. The outputs Z0 through Z11 are mapped through U40* and U41* to give the binary memory address needed by the RAM's.

With each memory cycle clock, the memory output data is strobed into U27* and U28* which provide the character code inputs to the ROM at U26*. The high order three bits CH7, CH6 and CH5 are used by the video generator section.

The operation of the cursor register has already been described in the previous section.

d. Video Generator Section:

The six bits of the ROM U26* output is dumped in parallel at end of each B5 into a PISO (parallel in serial out) shift register U22*. This shift register is serially shifted out at the BC rate to give the signal Q.

Mode register U46 stores the mode register setting upon a display control port write instruction with D7, D6 equal to 11. Signals CH7, CH6, CH5 and mode register outputs M1, M2, M4 are combined on card B to produce signals A-, B and C to control U28 for modulating or inverting the Q signal. U15 pin 8 provides an inhibit signal to suppress data read/write glitch and to provide a boarder around the display when inverted video (white background) display is selected. M3 inverts the video signal itself at U27 pin 6. U35 pin 12 takes into account the blanking signals. The 1/2 of U21 clocked at BC rate gives a glitch free video output VIDEO-. The VIDEO- and the composite sync are mixed by U24 and the M5 input inhibits the display.

3. KEYBOARD SECTION:

Provision has been made on VDB to service a parallel keyboard. DC power of +5v (upto 50 ma.) and -12v. (upto 250 ma.) may be drawn through the connector J2 to power the keyboard. (Caution: Keyboard requiring more DC than the above current limits may cause permanent damage to the voltage regulator circuits! External keyboard strobe signal KBS- is connected to pin 1 of J2 and is differentiated by U23* pin 6 to clock the keyboard data KB0 through KB7 into U5* and U6*. The differentiate strobe signal also presets 1/2 of U29* the status flip-flop, which is cleared by either POC· (power on clear) or keyboard read KBRD-. The keyboard control port when used as an output provides eight latched data lines via U11* and U12* for possible external use, as previously described.

For keyboards with ground-true data outputs, U1 and U2* must be replaced with 74368's unless a software complement is performed on the data read into the machine.

D. Checking Out The Assembled VDB

The VDB has been wave soldered, assembled and tested at the factory. However, before plugging it in, it should be inspected for any possible damage during shipping. Make sure all components are in place. Make sure the chips are properly oriented with reference to pin 1. Pin 1 is generally "UP" and to the right on both boards.

The next step is to insure the two boards are solidly mated together. Plug the boards into the system with power OFF. Hook up the video cable to the monitor.

In plugging the boards together, you will notice that connectors are oriented in such a way that the boards can only go together one way. Essentially it is with IC's pin 1 in "UP" direction on both boards.

*** W A R N I N G ***

*** DANGER *** DANGER *** DANGER ***

Connecting the VDB to a hot chassis television set, (i.e. 110 volts) will cause destruction of the ground traces on the PC boards, damage to IC's, and may even damage other PC boards in your computer.

Please be advised that this action will void the warranty and in this case TDL will not be responsible for damage to other equipment.

*** *** *** *** *** *** *** *** ***

With the 2 boards in the machine, video cable hooked up, and the machine OFF, the first thing to do is turn the video monitor ON and let it warm up. Adjust it for medium contrast, then turn the brightness control UP until you have a nice bright white screen called a RASTER. You then turn the brightness down until the raster just disappears. At this point you turn on the computer and you should get an indication on the screen of the Video. Perhaps the horizontal sync may be out or the vertical sync may be out. Adjust them and then lock them in. If you do not get any video indication, hit RESET. If this doesn't work measure the voltages using the schematic diagram as a reference. Check to see if the voltage regulator is properly functioning. There should be +8 volts going into the +5 volt voltage regulator and +5 volts coming out. (See tne power supply on CARD A Sheet 1 of 2 on the schematics.)

When the video is correct you should get many white boxes on the screen which essentially represent character positions. Depending on the garbage in the RAM memory when you power up, you may get miscellaneous characters scattered on the screen. Do not worry about them.

Sign on the Zapple Monitor, or load your usual operating system. Then load the appropriate VDB driver program. Three drivers will be supplied with the VDB. Two drive programs are for the Z-80 and the other driver is for the 8080. (A "driver" is a piece of software that interfaces a piece of hardware to an operating system.)

Recapping, load the appropriate VDB driver program into the machine, then assign the driver as one of the user defined output devices. (i.e. list device, console, punch, etc.) If, for example, the user defined output device is the console device, you have to have something as the input device. In many cases it may be a parallel keyboard interfaced to the VDB itself. (See the Keyboard documentation elsewhere in this manual.)

The Video portion of the VDB package may also be your television set with the proper adaption. (i.e. a modification of your television set). We highly recommend you buy a "Pickles & Trout" modification kit, which uses a black and white Hitachi television set. This allows direct video input. We DO NOT recommend using an R.F. converter, as the band width is not sufficient for the 80x25 display of the VDB.

The VDB screen should clear up on the first character output to it, and the cursor should be in the "Home" position.

This board has been fully tested at the factory. If you have a problem, please consult the troubleshooting guide, or return the board to the factory for repair. (See warranty at rear of manual)

E. VDB Keyboard Connections

The keyboard connector is J2 on card A of VDB. An Ansley 26-pin connector is used. Below are the pin assignments.

Pin	Signal
1	KBS- (input data strobe)
2	(not used) RESET
3	Ground
4	-5 v. DSP-
5	Ground
6	+12 v.
7	Ground
8	-12 v. 200 ma. maximum
9	KW2 (control output)
10	KW5 (control output)
11	KW6 (control output)
12	KW1 (control output)
13	KW7 (control output)
14	KW4 (control output)
15	KW0 (control output)
16	KW3 (control output)
17	KB2 (keyboard input)
18	KB1 (keyboard input)
19	KB5 (keyboard input)
20	KB6 (keyboard input)
21	KB4 (keyboard input)
22	KB7 (keyboard input)
23	KB3 (keyboard input)
24	KB0 (keyboard input)
25	+5 v. (50 ma. maximum)
26	+5 v. (50 ma. maximum) (Not OK)

F. Troubleshooting Hints

This section describes the procedure to properly troubleshoot a malfunctioning VDB. The minimum equipment required is a 15 mHz. oscilloscope, a VOM, and perhaps a digital logic probe would be useful. Some knowledge of digital circuitry is also required.

1. Inspect for obvious faults, i.e. broken leads on crystal or other discrete components, metal filings and solder splashes. Also, check for IC leads missing their socket holes or bent underneath the IC bodies.

2. Remove All IC's from both boards.

3. Check continuity of all DC supply voltages on all IC sockets including ground with an ohmmeter. Do this with Card A and Card B mated.

4. With power OFF insert card in slot on an extender, make sure the card orientation is correct, or else permanent damage may result.

5. Turn power on, check DC voltages at output of regulators VR1, VR3 (+5v); VR2 (+12v) and anode side of D1 (-5v) and D2 (-12v). Any reading deviating from nominal value by more than +/- 10% may indicate a short, defective regulator/zener or a reversed capacitor or diode.

6. With power off, remove VDB:
   Load on Card A: U17
   Load on Card B: U7,U8,U9,U10,U11,U12,U13,U14,U15,
   U17,U18,U19,U20,
   U23,U24,U25,U26
   U29
   U37

Connect a jumper next to U11 from J to D. Insert the VDB in, connect the video cable from the TV or monitor to the VDB video output. Power on. If you can sync your TV (a blank picture), most of the timing signals are then okay. If you can't sync the TV, you need an oscilloscope to locate the fault. Check in the following order:

a. Card A: U17 pin 4 and pin 6. High frequency 11.36916 mHz clock This signal means crystal oscillator is okay.

b. Card B: U13 pin 2. Positive narrow clock at 1.89486 mHz. This signal means dot counter (U20) is okay.

c. Card B: U13 pin 4. Positive narrow clock at 15.66 kHz. This means x-counter (U10,U12) is okay.

d. Card B: U13 pin 12. Positive narrow clock at 1.74 kHz. This means video slice counter (U19) is okay.

e. Card B: U25 pin 10. Rectangular wave at 60 Hz. This means the Y-counter (U7,U8) is okay.

f. Card B: U25 pin 8. Square wave at 3.75 Hz. This means the blink counter (U25) is okay.

Your computer should function normally since only DC is taken from the computer and no signal interchange has taken place.

7. Load Card A: U7, U8, U11, U12, U18, U19, U24, U25, U29 Card B U39. Your S-100 Bus computer should function normally with power on. These IC's receive bus signals only and no signal is yet fed into the S-100 bus from the VDB. If it is not functioning, check the IC's for misplacement, misorientation or shorted inputs.

8. Load all IC's EXCEPT the following:

Card A: U1, U2, U3, U4 Card B: U43, U47, U48, U50

Set up the port address switch according to Table 1 on Page 24. The computer should function normally with the VDB in as now populated. All IC's which feed signals into the bus have yet to be inserted.

On the TV monitor you should see random data. If your machine has a power on clear or reset, turning power on or pushing reset should give you a blinking cursor at the upper left hand corner of the screen.

If there is no picture check the following locations:

a. Card A: U38 pin 5. A 1.89486 mHz clock at 0v to +12v amplitude. This is the dynamic memories' clock.

b. Card A: U27 pins 4,5,12,13. U28 pins 4,5,12,13. These are memory outputs. Changing at memory clock rate.

c. Card A: U27, U28 pins 2,7,10,15. These are memory output buffer outputs. Changing at memory clock rate.

d. Card A: U26 pins 5,6,7,17,18,19. These are ROM outputs changing at memory clock rate.

e. Card A: U22 pin 13. This is parallel-to-serial shift reg. output, changing at crystal clock oscillator rate.

f. Card B: U28 pin 5.
This is partially modified raw video signal to account for cursor, blinking frame edging and graphic symbol inversion. Changing at crystal clock rate. U28 pin 7 must not be high all the time.

g. Card B: U27 pin 6, then U35 pin 12.
Both changing at crystal clock oscillator rate.

h. Card B: U24 pin 3.
A cleaned up version of the final video signal. U24 pin 5 should be high. If not, remove U24 bend out pin 6, reinsert. (We have to go back to check the mode register reset later.)

9. Insert all remaining IC's. If the computer functions normally, you should try loading the VDB driver and use the program to establish that the VDB functions properly. If the computer does not function normally, remove Card A U1,U2,U3,U4 and Card B U43, U47,U48,U50 and proceed with program test as follows: (The I/O ports are assumed to be E0,E1,E2 and E3)

a. Port Selection Test: Negative pulses should be found at U21.

```
TEST1: OUT E0   ;Card A U21 pin 9
       OUT E1   ;Card A U21 pin 10
       OUT E2   ;Card A U21 pin 11

       IN  E0   ;Card A U21 pin 7
       IN  E1   ;Card A U21 pin 6
       IN  E2   ;Card A U21 pin 4
       IN  E3   ;Card A U21 pin 5
       JMP TEST1
```

Total absense of signal would suggest wrong DIP switch setting.

b. X,Y cursor and Mode Register Test:

```
TEST2: MVI A,0
       OUT E0
       MVI A,0BFH
       OUT E0          ;Clear Mode register
       MVI A,0D0H
       OUT E0
       JMP TEST2
```

Pulses should appear on Card B
U44, U51 pin 2 (X-cursor)
U42, U49 pin2 (Y-cursor),
U46 pin 9. (Mode register)
and on Card B U51 pins 11,12,13,14, U44 pins 12,13,14
U49, pins 11,12,13,14, U42 pin 14,
U46 pins 2,5,7,10,12,15
should show square waves with the following program

running.

```
TEST3: MVI A,0
       OUT E0
       MVI A,80H
       OUT E0
       MVI A,0C0H
       OUT E0

       MVI A,7FH
       OUT E0
       MVI A,0BFH
       OUT E0
       MVI A,0DFH
       OUT E0
       JMP TEST3
```

The following program will test the reset function on these registers:

```
TEST4: MVI A,7FH
       OUT E0
       MVI A,0BFH
       OUT E0
       MVI A,0DFH
       OUT E0
       MVI A,0FFH
       OUT E0
       JMP TEST4
```

Check pin 1's of U42,U44,U46,U49,U51 for reset pulse, and the register outputs for rectangular waves.

G. Using The VDB Driver Software

This driver is supplied in three formats:

1. 8080 version. -Requires 1/4K ROM or RAM, and 80 Bytes of RAM only. (Driver only, no graphics.)
2. Z-80 ROM version. -Requires 1/4K ROM, no RAM needed.
3. Z-80 RAM version. -Requires 1/4K RAM/ROM, 80 Bytes RAM.

The 80 Bytes buffer areas are used to speed up the transfer of data to and from the VDB. These may be placed anywhere by changing the "LXI H,BUFF" located at VDBCO+14H. (See listing.)

The RAM versions are recommended, as they are faster. The ROM version is supplied for those who might need a driver which does not depend on any RAM memory for its operation.

For those who wish to use the Z-80 drivers, but do not have Zap or Zapple, they will have to either use the 8080 versions or HAND LOAD the Z-80 versions. A complete listing is provided for this purpose.

This driver will simulate the normal teleprinter/dumb terminal style of operation, with SCROLLING, and also will recognize certain control characters. The following is a list of the control characters and the effect they have on the screen:

CHARACTER	HEX	MEANING
back-space	08H	Moves cursor to left
Line-feed	0AH	Moves cursor down, scrolls if at bottom
Form-feed	0CH	Clears screen, homes cursor.
Carriage ret.	0DH	Returns cursor to left margin.
DC-1	11H	Switches pages
DC-2	12H	Unconditionally SETS or RESETS bit 7.
DC-3	13H	Enables/disables blink.
DC-4	14H	Switches between standard and reverse video.

The VDB driver is "CALLed" with the character to be acted upon in the "C" register. It conforms to ZAPPLE configurations. It may be "Assigned" as the User-defined LIST device, by patching a "JMP" in the proper 0F800H vector

to the driver's location. It may be both the input (parallel keyboard) and output device for a User defined CONSOLE device. Three locations must be patched in this case. (CI, CO, CSTS). The keyboard input software is included in the driver, but may be loaded elswhere. It is a short routine, not unlike using a UART. The TEST BIT is bit 7. It may be optioned for TRUE or FALSE using the "ABC" jumpers. It is sent assuming the "A to B" configuration (TRUE), i.e. if upon reading the Keyboard Status, bit 7 is low (0), then there has not been a keyboard strobe since the data port was last read. If it is high (1), then there IS data waiting. Reading the data port will clear the high, thus readying it for the next keyboard strobe.

A note about the self-initializing feature. The reason for this would be if the main system were just loaded from power-on condition, there would be garbage on the VDB screen. When the software is loaded (assuming RAM now), the very first time the VDB is accessed, it will first clear the screen. This simplifies power-up starting. Another way (as with ROM) is to simply first send a Form-feed before sending any other characters to the VDB.

HARDWARE NOTES:

There are four ports in the VDB. The first (lowest) is the VDB control. The next is the VDB data. The third is KYBD status, and the fourth is KYBD data.

Before reading the VDB control port, the port must be written with various masks to determine what information will be present when the control port is read. The following table demonstrates this:

MASK	MEANING
11100000	Read current X position
11100001	Read current Y position
11100010	Read current MODE register
11100011	RESET VDB. (Software reset.)

When writing to the control port, the upper 2 bits (6,7) determine what is being written. See the table below:

MASK	MEANING
00XXXXXX	Write X cursor position (0 thru 79)
11XXXXXX	Write Y cursor position (0 thru 24)
10XXXXXX	Write a new value to the MODE register

When reading the MODE register, the following bits have the following meanings:

MASK	MEANING
XXXXXXX1	Alternate page flag
XXXXXX1X	Blink is enabled
XXXXX1XX	Characters are inverted
XXXX1XXX	Display is inverted
XXX1XXXX	Enable/disable cursor
XX1XXXXX	For blanking entire display

H. VDB Parts List

1. Card A

1.1 Integrated Circuits

8--4050	U30 through U37
3--74LS00	U14, U15, U23
3--74LS04	U16, U17, U39
1--74LS10	U25
1--74LS14	U24
2--74LS74	U13, U29
1--74LS139	U21
2--74LS157	U40, U41
1--74LS166	U22
2--74173	U3, U4
2--74LS173	U9, U10
6--74LS175	U5*, U6*, U11*, U12*, U27, U28
2--74LS266	U18, U19
4--74367	U1*, U2*, U7, U8
2--7805, +5 v. regulator, 1a.	VR1, VR3
1--7812, +12 v. regulator, 1a.	VR2
1--DS3672N, clock driver	U38
1--MCM6570, custom masked ROM	U26

(Note: Items with * not needed if keyboard interface not desired.)

1.2 Diodes

1--1N751, Zener	D1
1--1N4742, Zener	D2

1.3 Resistors, 1/4 w. 10% tolerance unless otherwise stated.

1--10 ohm	R22
3--100 ohm	R1, R13, R25
1--150 ohm, 1/2 w.	R24
2--390 ohm	R2, R3
1--1k ohm	R7
1--2.2k ohm	R23
8--2.7k ohm	R14 through R21
1--3.9k ohm	R11
6--4.7k ohm	R4, R5, R6, R8, R9, R10
1--5.6k ohm	R12

1.4 Capacitors

1--10 pf	C11
1--220 pf	C25
3--470 pf	C5, C16, C44
33--.1 uf ceramic bypass	C1 through C4, C6 through C10, C12 throught C15 C17 through C24, C26 through C37

5--33 uf tantalum tear-drop, 25 v. C38-C40, C42,C43

1.5 Other Items

1--printed circuit card, VDB card A
3--heat sink, ***
1--8-pin IC socket
12--14-pin IC socket
18--16-pin IC socket
8--18-pin IC socket
1--24-pin IC socket
1--DIP switch, 8PST U20
1--Molex 09-52-3102 S2
3--Molex 09-52-3152 S1, S3
1--Ansley 609-2602M J2
1--Crystal, General Purpose, Series Mode, 11.36916 MHz., HC-18/U holder
1--Keyboard cable assembly, ***(identical to SMB part)
3--6-32, 1/4" panhead screw
3--#6 star washer
3--6-32 nut
1--6"-length insulating tape

2. Card B

2.1 Integrated Circuits

4--74LS00	U14, U18, U31, U39
1--74LS02	U29
4--74LS04	U13, U26, U40, U52
1--7406	U24
1--74LS08	U37
3--74LS10	U15, U35, U45
2--74LS20	U9, U41
2--74LS32	U36, U38
1--7472	U30
2--74LS74	U21, U32
1--74LS86	U27
3--74LS107	U7, U17, U22
1--74LS139	U33
1--74LS151	U28
3--74LS157	U1, U3, U5
9--74LS163	U8, U10, U12, U19, U20, U42, U44, U49, U51
1--74LS164	U11
2--74LS174	U16, U46
2--74LS175	U23, U34
3--74LS266	U2, U4, U6
1--74LS293	U25
4--74367	U43, U47, U48, U50

2.2 Resistors, 1/4 w., 10% tolerance

1--20 ohm	R4
1--75 ohm	R5
2--100 ohm	R6, R7
1--200 ohm	R3
1--330 ohm	R2
1--1k ohm	R1

2.3 Capacitors

2--470 pf	C48, C49
2--33 mf, 5 v. electrolytics,	C46, C47
45--.1 uf ceramic bypass	C1 through C45

2.4 Other Items

1--printed circuit card, VDB card B	
30--14-pin IC socket	
22--16-pin IC socket	
1--Molex 09-66-1031	
1--Molex 09-64-1101	P2
3--Molex 09-64-1151	P1, P3, P4
1--Molex 09-50-7031	(for video connection)
3--Molex 08-50-0106 terminal	
9--Augat pin, ***	

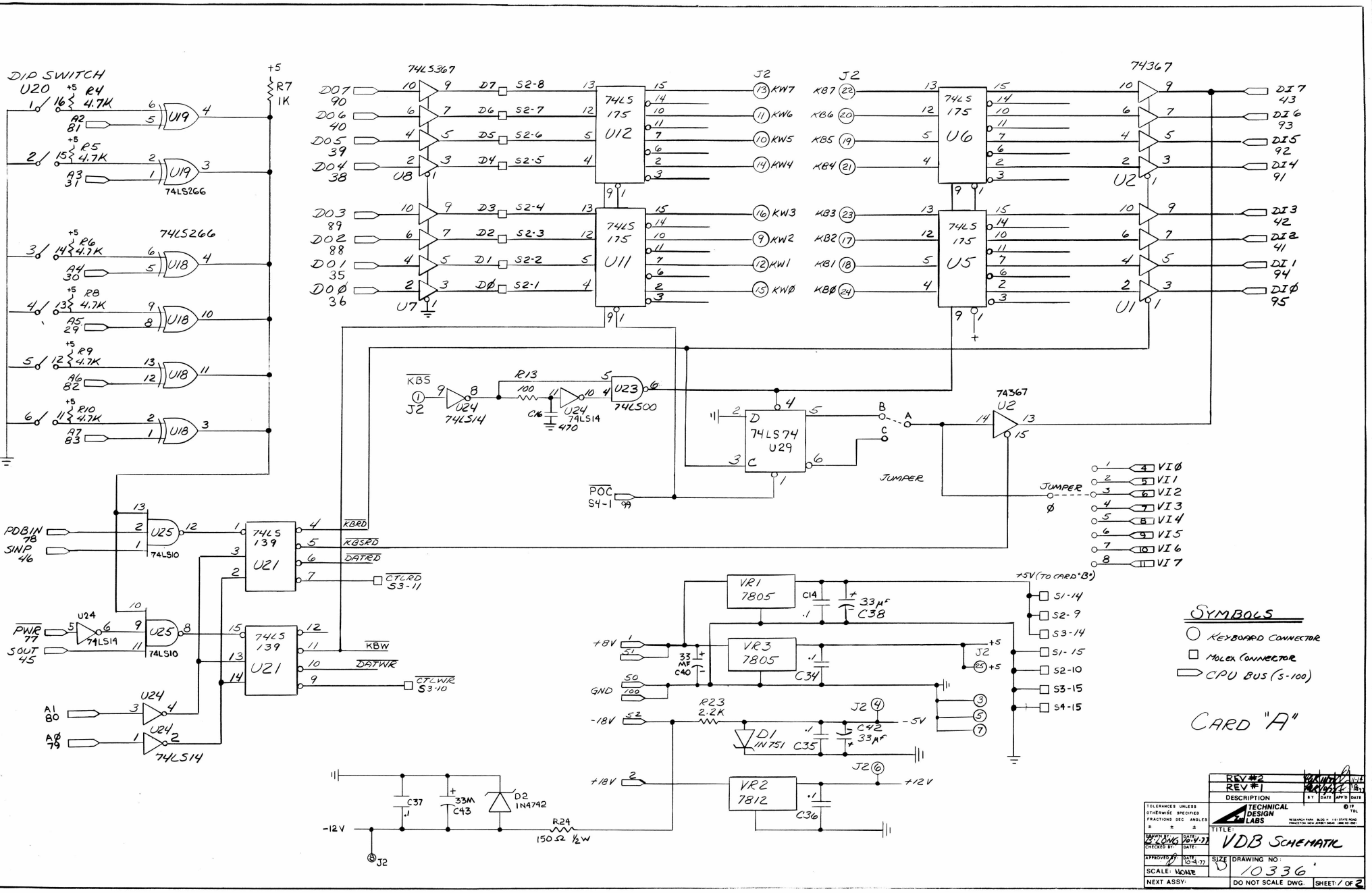
Table 1: VDB Port Addresses

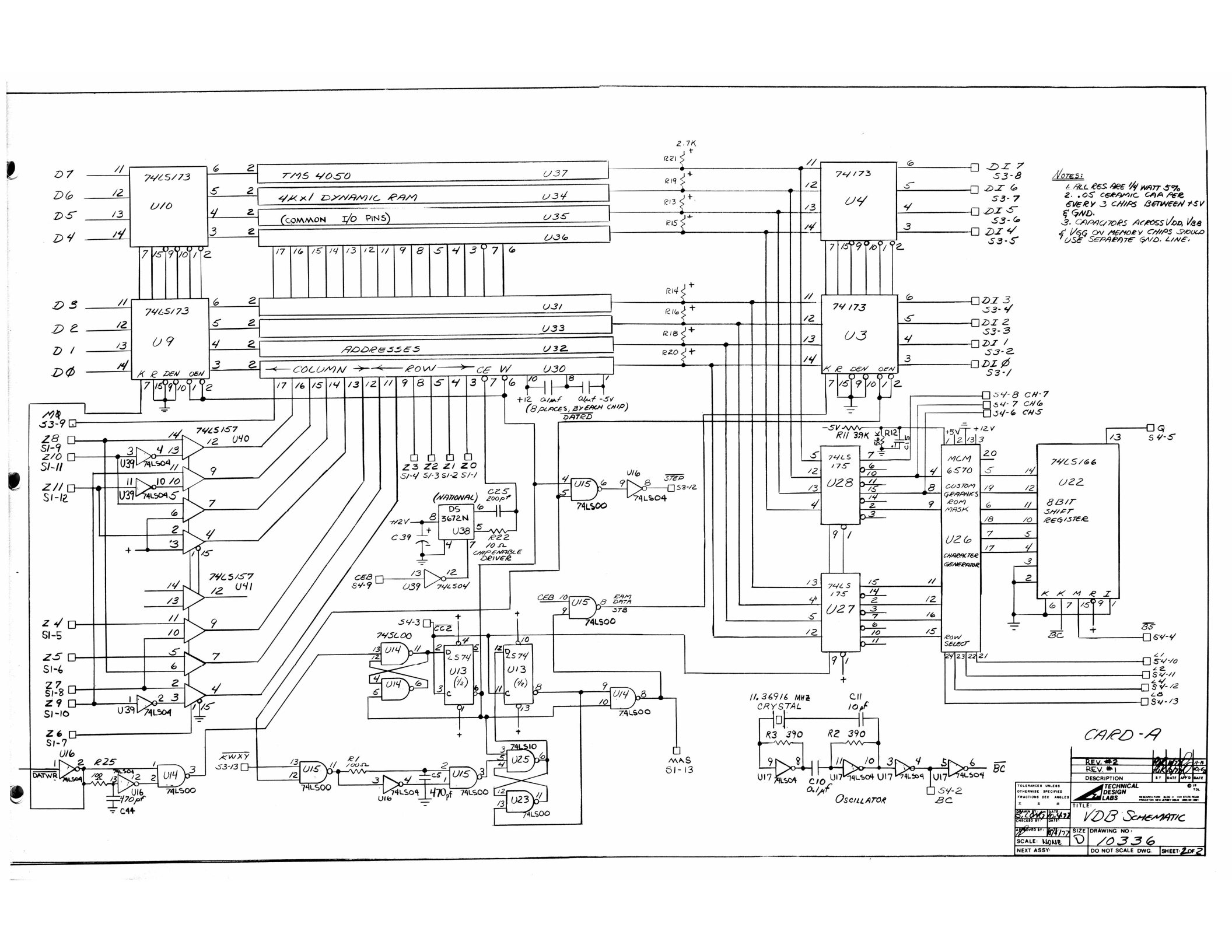
U20/A=Switch, OFF=1=top depressed ON=0=bottom depressed						Display Control Port	Display Data Port	Keyboard Control Port	Keyboard Data Port	
Switch # 6	5	4	3	2	1					
A7	A6	A5	A4	A3	A2	ØØ	Ø1	1Ø	11	← A1 AØ
Ø	Ø	Ø	Ø	Ø	Ø	ØØ	Ø1	Ø2	Ø3	
Ø	Ø	Ø	Ø	Ø	1	Ø4	Ø5	Ø6	Ø7	
Ø	Ø	Ø	Ø	1	Ø	Ø8	Ø9	ØA	ØB	
Ø	Ø	Ø	Ø	1	1	ØC	ØD	ØE	ØF	
Ø	Ø	Ø	1	Ø	Ø	1Ø	11	12	13	
Ø	Ø	Ø	1	Ø	1	14	15	16	17	
Ø	Ø	Ø	1	1	Ø	18	19	1A	1B	
Ø	Ø	Ø	1	1	1	1C	1D	1E	1F	
Ø	Ø	1	Ø	Ø	Ø	2Ø	21	22	23	
Ø	Ø	1	Ø	Ø	1	24	25	26	27	
Ø	Ø	1	Ø	1	Ø	28	29	2A	2B	
Ø	Ø	1	Ø	1	1	2C	2D	2E	2F	
Ø	Ø	1	1	Ø	Ø	3Ø	31	32	33	
Ø	Ø	1	1	Ø	1	34	35	36	37	
Ø	Ø	1	1	1	Ø	38	39	3A	3B	
Ø	Ø	1	1	1	1	3C	3D	3E	3F	
Ø	1	Ø	Ø	Ø	Ø	4Ø	41	42	43	
Ø	1	Ø	Ø	Ø	1	44	45	46	47	
Ø	1	Ø	Ø	1	Ø	48	49	4A	4B	
Ø	1	Ø	Ø	1	1	4C	4D	4E	4F	
Ø	1	Ø	1	Ø	Ø	5Ø	51	52	53	
Ø	1	Ø	1	Ø	1	54	55	56	57	
Ø	1	Ø	1	1	Ø	58	59	5A	5B	
Ø	1	Ø	1	1	1	5C	5D	5E	5F	
Ø	1	1	Ø	Ø	Ø	6Ø	61	62	63	
Ø	1	1	Ø	Ø	1	64	65	66	67	
Ø	1	1	Ø	1	Ø	68	69	6A	6B	
Ø	1	1	Ø	1	1	6C	6D	6E	6F	
Ø	1	1	1	Ø	Ø	70	71	72	73	
Ø	1	1	1	Ø	1	74	75	76	77	
Ø	1	1	1	1	Ø	78	79	7A	7B	
Ø	1	1	1	1	1	7C	7D	7E	7F	

Table 1 - cont.

VDB Port Addresses

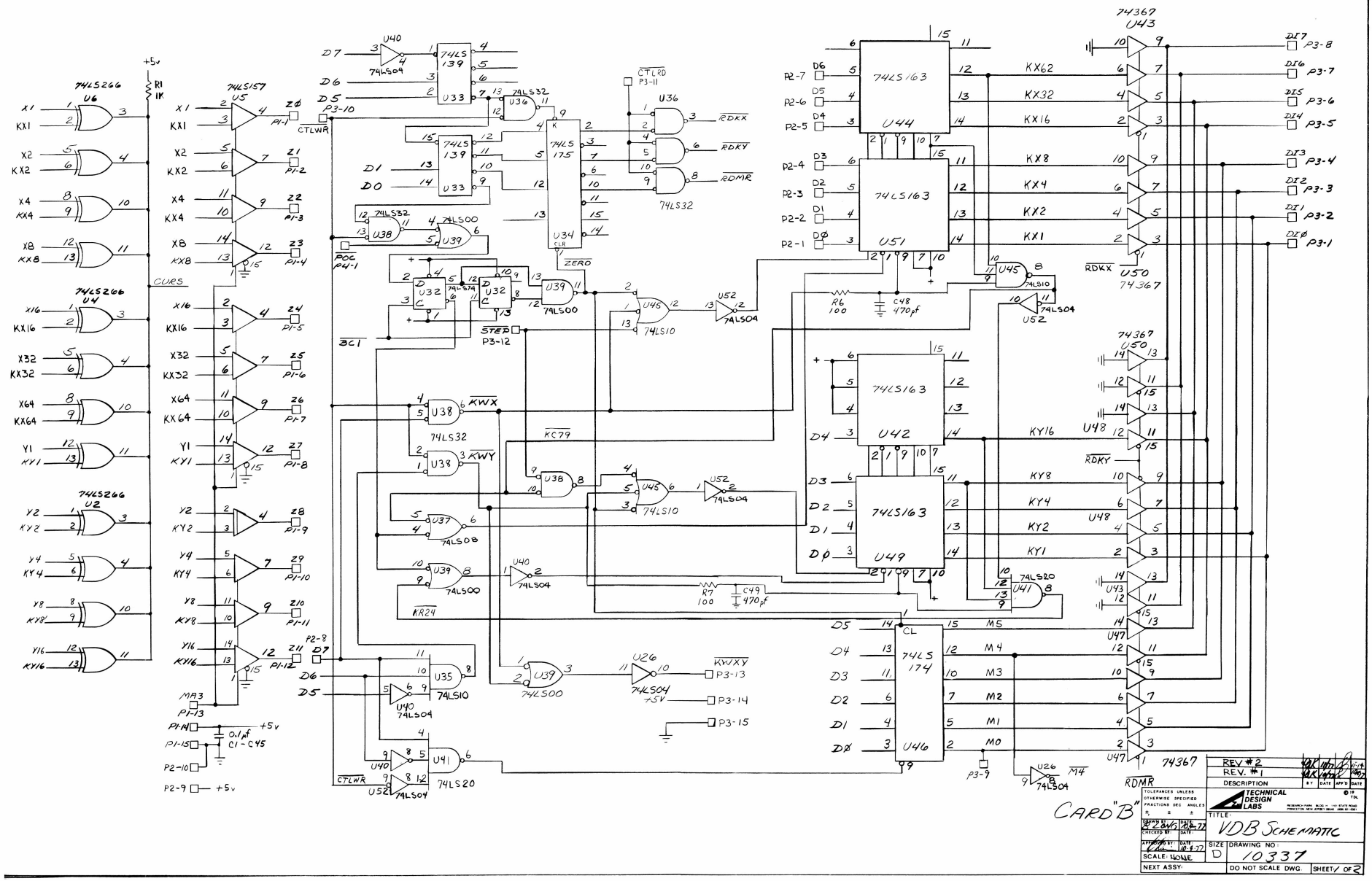
U20/A=Switch, OFF=1=top depressed ON=0=bottom depressed						Display Control Port	Display Data Port	Keyboard Control Port	Keyboard Data Port	
Switch # 6	5	4	3	2	1					
A7	A6	A5	A4	A3	A2	ØØ	Ø1	1Ø	11	←A1 AØ
1	Ø	Ø	Ø	Ø	Ø	8Ø	81	82	83	
1	Ø	Ø	Ø	Ø	1	84	85	86	87	
1	Ø	Ø	Ø	1	Ø	88	89	8A	8B	
1	Ø	Ø	Ø	1	1	8C	8D	8E	8F	
1	Ø	Ø	1	Ø	Ø	9Ø	91	92	93	
1	Ø	Ø	1	Ø	1	94	95	96	97	
1	Ø	Ø	1	1	Ø	98	99	9A	9B	
1	Ø	Ø	1	1	1	9C	9D	9E	9F	
1	Ø	1	Ø	Ø	Ø	AØ	A1	A2	A3	
1	Ø	1	Ø	Ø	1	A4	A5	A6	A7	
1	Ø	1	Ø	1	Ø	A8	A9	AA	AB	
1	Ø	1	Ø	1	1	AC	AD	AE	AF	
1	Ø	1	1	Ø	Ø	BØ	B1	B2	B3	
1	Ø	1	1	Ø	1	B4	B5	B6	B7	
1	Ø	1	1	1	Ø	B8	B9	BA	BB	
1	Ø	1	1	1	1	BC	BD	BE	BF	
1	1	Ø	Ø	Ø	Ø	CØ	C1	C2	C3	
1	1	Ø	Ø	Ø	1	C4	C5	C6	C7	
1	1	Ø	Ø	1	Ø	C8	C9	CA	CB	
1	1	Ø	Ø	1	1	CC	CD	CE	CF	
1	1	Ø	1	Ø	Ø	DØ	D1	D2	D3	
1	1	Ø	1	Ø	1	D4	D5	D6	D7	
1	1	Ø	1	1	Ø	D8	D9	DA	DB	
1	1	Ø	1	1	1	DC	DD	DE	DF	
1	1	1	Ø	Ø	Ø	EØ	E1	E2	E3	
1	1	1	Ø	Ø	1	E4	E5	E6	E7	
1	1	1	Ø	1	Ø	E8	E9	EA	EB	
1	1	1	Ø	1	1	EC	ED	EE	EF	
1	1	1	1	Ø	Ø	FØ	F1	F2	F3	
1	1	1	1	Ø	1	F4	F5	F6	F7	
1	1	1	1	1	Ø	F8	F9	FA	FB	
1	1	1	1	1	1	FC	FD	FE	FF	

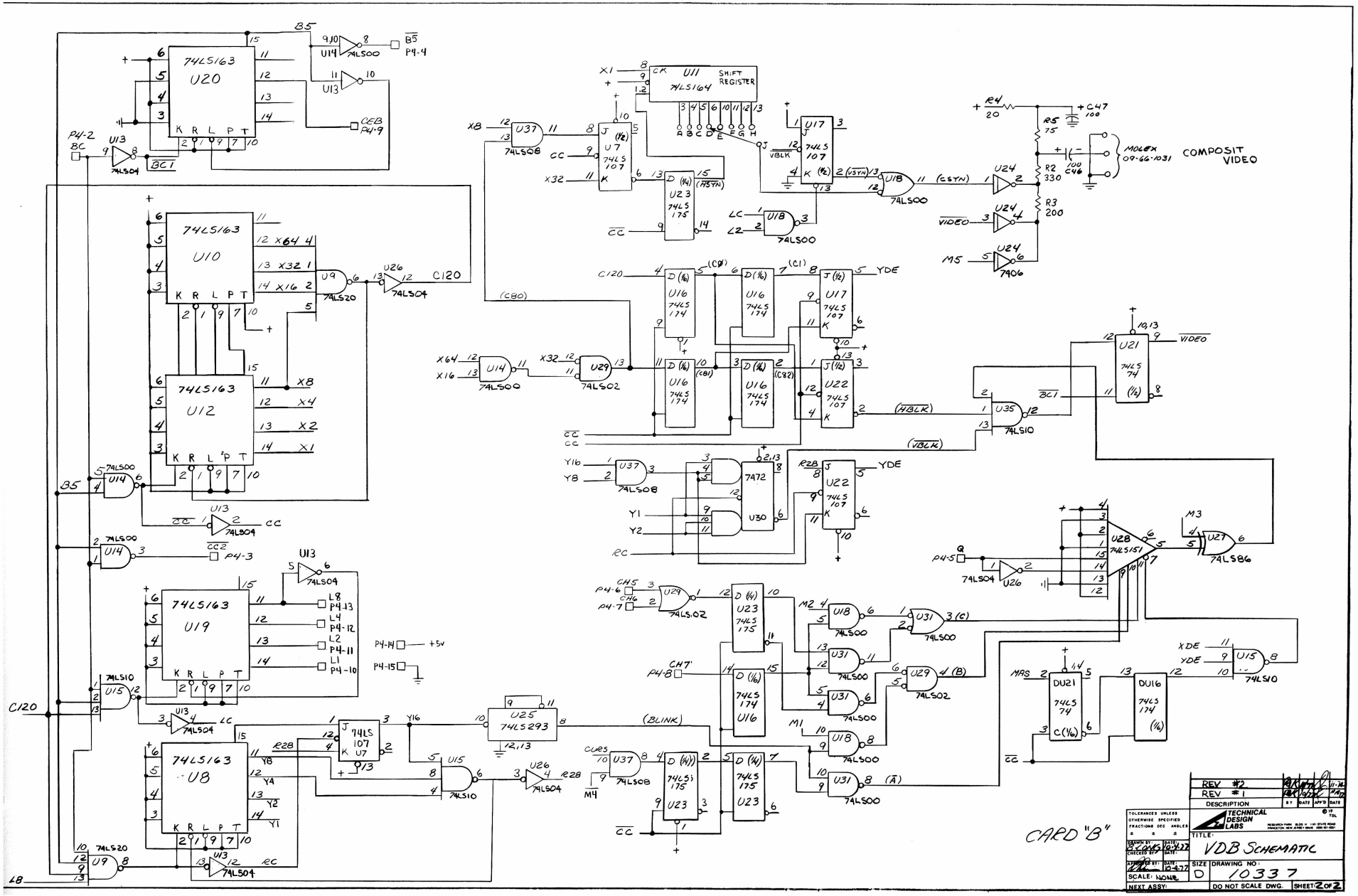

DIP SWITCH
U20
74LS367
74LS266
74LS139
U21
U12
U11
U8
U7
U6
U5
U2
U1
74LS175
74LS74
U29
74LS500
74LS14
74LS10
74LS19
VR1 7805
VR2 7812
VR3 7805
D1 IN751
D2 IN4742
R24 150Ω ½W
R23 2.2K
C37 .1
33M C43
C38 33MF
C40 33MF
C42 33MF
C34 .1
C35 .1
C36 .1
C14 .1
C4 470
R13 100
+5V (TO CARD "B")
-5V
+12V
-12V
+8V
-18V
GND
JUMPER
SYMBOLS
KEYBOARD CONNECTOR
MOLEX CONNECTOR
CPU BUS (S-100)
CARD "A"
VDB SCHEMATIC
TECHNICAL DESIGN LABS
DRAWING NO: 10336
SHEET 1 OF 2
DO NOnot SCALE DWG.

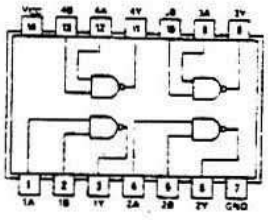


NOTES:
1. ALL RES. ARE 1/4 WATT 5%
2. .05 CERAMIC CAP PER EVERY 3 CHIPS BETWEEN +5V & GND.
3. CAPACITORS ACROSS VDD, VBB & VGG ON MEMORY CHIPS SHOULD USE SEPARATE GND. LINE.

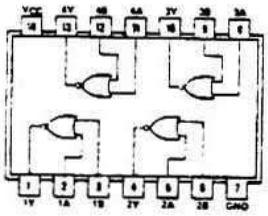
REV. #2	
REV. #1	
DESCRIPTION	BY / DATE / APP'D / DATE
TECHNICAL DESIGN LABS	© 19 TDL
TITLE: VDB SCHEMATIC	
SIZE: D	DRAWING NO: 10336
SCALE: NONE	DO NOT SCALE DWG.
NEXT ASSY:	SHEET: 2 OF 2


74LS266 U6
74LS157 U5
74LS266 U4
74LS266 U2
74LS40 U40
74LS139 U33
74LS32 U36
74LS175 U34
74LS32
74LS32 U38
74LS00 U39
74LS74 U32
74LS10 U45
74LS04 U52
74LS08 U37
74LS10 U35
74LS04 U40
74LS20 U41
74LS04 U26
74LS163 U44
74LS163 U51
74LS163 U42
74LS163 U49
74LS174 U46
74367 U43
74367 U50
74367 U48
74367 U47
R1 1K
R6 100
C48 470pf
R7 100
C49 470pf
0.1mf C1-C45
CTLWR
CTLRD P3-11
STEP P3-12
ZERO
RDKX
RDKY
RDMR
KWX
KWY
KWXY
KC79
KR24
CURS
MA13
CARD "B"
REV #2
REV. #1
DESCRIPTION
TECHNICAL DESIGN LABS
TITLE: VDB SCHEMATIC
SIZE D
DRAWING NO: 10337
SCALE: NONE
NEXT ASSY:
DO NOT SCALE DWG.
SHEET 1 OF 2

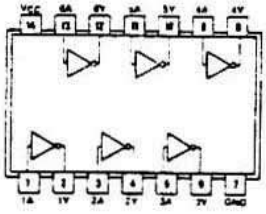

74LS163 U20
74LS163 U10
74LS163 U12
74LS163 U19
74LS163 U8
U11 SHIFT REGISTER 74LS164
U7 74LS 107
U23 74LS 175
U16 74LS 174
U17 74LS 107
U22 74LS 107
U21 74LS 74
U28 74LS151
U25 74LS293
U30 7472
MOLEX 09-66-1031
COMPOSIT VIDEO
CARD "B"
REV #2
REV #1
DESCRIPTION
TECHNICAL DESIGN LABS
TITLE: VDB SCHEMATIC
SIZE D
DRAWING NO: 10337
SCALE: NONE
SHEET 2 OF 2
DO NOT SCALE DWG.



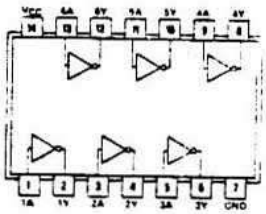
74LS00



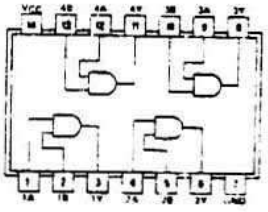
74LS02



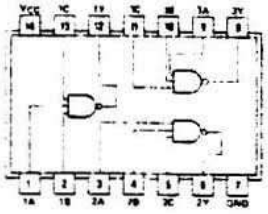
74LS04



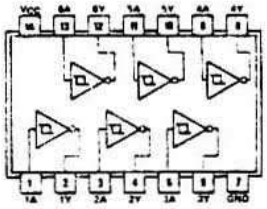
7406



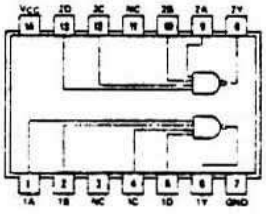
74LS08



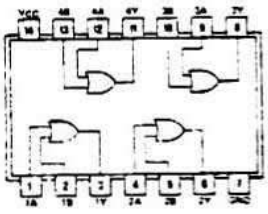
74LS10



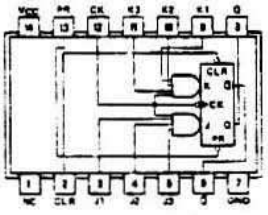
74LS14



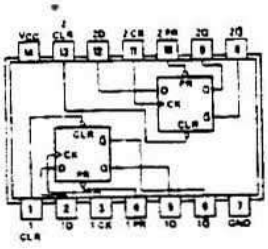
74LS20



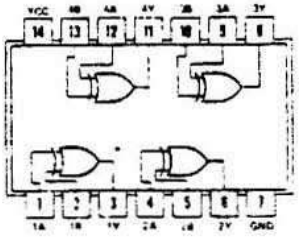
74LS32



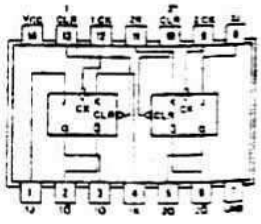
7472



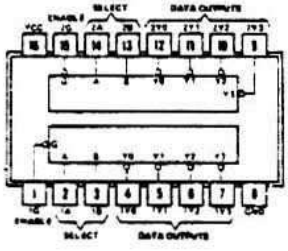
74LS74



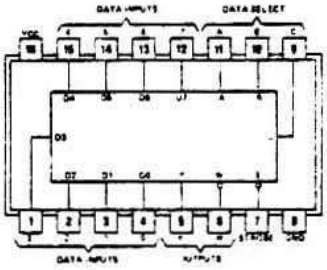
74LS86



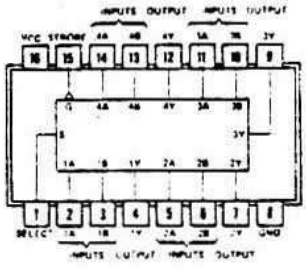
74LS107



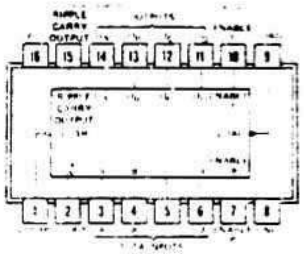
74LS139



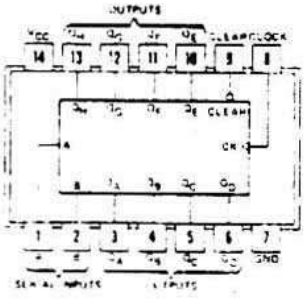
74LS151



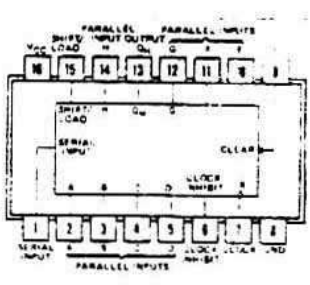
74LS157



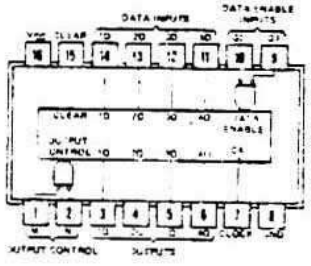
74LS163



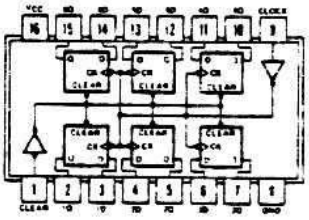
74LS164



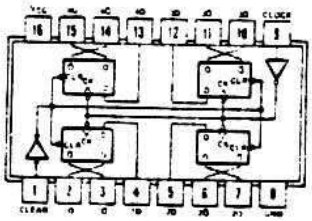
74LS166



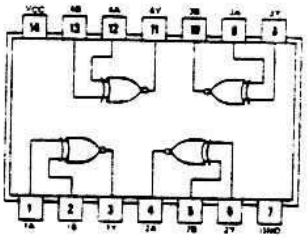
74LS173, 74173



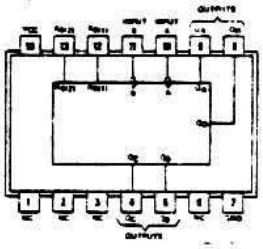
74LS174



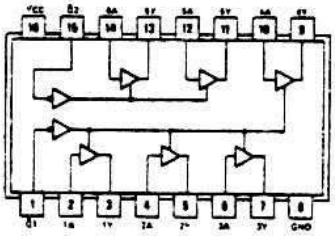
74LS175



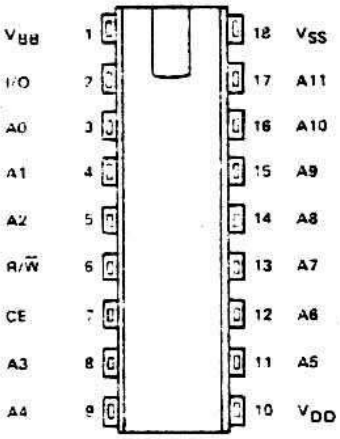
74LS266



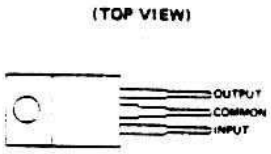
74LS293



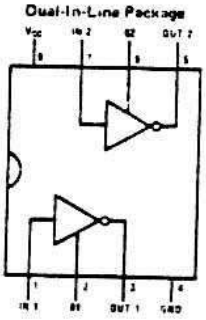
74367



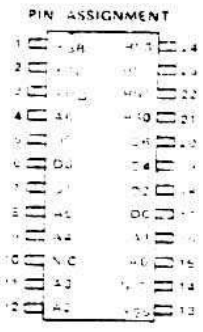
4050



7805, 7812



DS3672N



MCM6570

4050 & 7400 series pinouts courtesy of Texas Instruments.

```
                        ;
                                .REMARK "

                        THIS DRIVER IS DESIGNED TO INTERFACE THE
                        TDL ZAPPLE MONITOR WITH THE TDL VIDEO
                        DISPLAY BOARD (VDB). IT PROVIDES BOTH
                        A CHARACTER ORIENTED (TELETYPE-LIKE)
                        INTERFACE AND A X/Y PLOT INTERFACE.

                        Variable parameters are obtained at assembly
                        time. The RAM version buffer may be anywhere
                        in memory, and requires 80 decimal bytes.
                        The RAM program itself may be put in ROM,
                        but the self initializing feature must be
                        disabled.
                        "
                        ;
FC00                    BUF.    =\      "BUFFER LOCATION?"
                        ;
                        ;
                                .IFN    BUF.,[
FC00                    BUFF    =       BUF.
                        ]
                                .IFE    BUF.,[
                        BUFF    =       END.
                        ]
                        ;
                        .XLINK
                                .PABS
F900                    LOC     =\      "ENTER STARTING ADDRESS"
                        ;
                        ;
                        .I8080  ;8080 VERSION
                        ;
                        ;
                        .PHEX
                        ;
                        ; COPYRIGHT 1977 By Technical Design Labs, Inc.
                        ; 1101 State Rd. Research Park, Bldg. H
                        ; Princeton, New Jersey  08540
                        ;
```

```
                        ;
                        ;
F800                    .LOC    0F800H          ;INSERT JMPS FOR USER CONSOLE
                        ;
F800    C3 FA0C         JMP     KBIN
F803    C3 F900         JMP     VDBCO
                        ;
F818                    .LOC    0F818H          ;PLUG CSTS
                        ;
F818    C3 FA16         JMP     KBSTS
                        ;
```

```
                        ;
                        ;
                        ;       +++ VDB PORT AND MASK DEFINITIONS +++
                        ;
00E0                    VDBCTL  =\      "ENTER VDB STATUSPORT ADDRESS"
00E1                    VDBDAT  ==      VDBCTL+1        ;DATA PORT ADDRESS
00E2                    VDBK.S  ==      VDBCTL+2        ;VDB KEYBOARD STATUS
00E3                    VDBK.D  ==      VDBCTL+3        ;KEYBOARD DATA
0080                    VDBMRF  ==      10000000B       ;MODE REGISTER FLAG
00C0                    VDBYCF  ==      11000000B       ;Y CURSOR FLAG
00E0                    VDBXCR  ==      11100000B       ;X CURSOR READ
00E1                    VDBYCR  ==      11100001B       ;Y CURSOR READ
00E2                    VDBMRR  ==      11100010B       ;MODE REGISTER READ
00E3                    VDBRES  ==      11100011B       ;RESET VDB
0060                    VDBGMK  ==      01100000B       ;GRAPHIC MASK
                        ;
                        ;       +++ MODE BIT DEFINITIONS +++
                        ;
0000                    VDB%AP  ==      0               ;ALTERNATE PAGE
0001                    VDB%BE  ==      1               ;BLINK ENABLE
0002                    VDB%IS  ==      2               ;INVERT SYMBOL
0003                    VDB%ID  ==      3               ;INVERT DISPLAY
0004                    VDB%DC  ==      4               ;DISABLE CURSOR
0005                    VDB%DD  ==      5               ;DISABLE DISPLAY
                        ;
                        ;       +++ ZAPPLE MONITOR EQUATES +++
                        ;
F000                    ZAPPLE  ==      0F000H          ;ZAPPLE MONITOR LOCATIO

                        ;
                        ;       +++ ASCII CONTROL CHARACTERS +++
                        ;
0008                    BS      ==      08H             ;BACK SPACE
0009                    HT      ==      09H             ;TAB
000D                    CR      ==      0DH             ;CARRIAGE RETURN
000A                    LF      ==      0AH             ;LINE FEED
000C                    FF      ==      0CH             ;FORM FEED
0007                    BEL     ==      07H             ;BELL
0011                    DC1     ==      11H             ;DC CONTROLS
007F                    DEL     ==      7FH             ;DELETE
                        ;
```

RAM/8080


```
                        ;
                        ;       +++ VDB TELETYPE SIMULATOR +++
                        ;
F900                            .LOC    LOC
                        ;
F900    C5              VDBCO:  PUSH    B               ;SAVE REGISTERS
F901    D5                      PUSH    D               ;
F902    E5                      PUSH    H               ;
F903    CD F9FC         MAIN:   CALL    INIT            ;CALL DRIVER
F906    E1                      POP     H               ;RESTORE REGISTERS
F907    D1                      POP     D               ;
F908    C1                      POP     B               ;
F909    79                      MOV     A,C             ;OUTPUT CHARACTER INTO A

F90A    C9                      RET                     ;DONE
                        ;
                        ;
                        ;
                        ;       +++ RAM VDB DRIVER +++
                        ;
F90B    79              VDB:    MOV     A,C             ;GET OUTPUT CHARACTER
F90C    E67F                    ANI     7FH             ;GET RID OF PARITY BIT
F90E    C8                      RZ                      ;IGNORE NULLS
F90F    FE7F                    CPI     DEL             ;RUBOUT?
F911    C8                      RZ                      ;IGNORE IT
F912    0EE0                    MVI     C,VDBCTL        ;SET C UP
F914    21 FC00                 LXI     H,BUFF          ;POINT TO BUFFER
F917    FE20                    CPI     " "     ;CONTROL CHARACTER?
F919    DA F97D                 JC      CNTL    ;YES
F91C    47                      MOV     B,A     ;SAVE PRINTING CHAR.
F91D    CD F9E4                 CALL    GTMD    ;GET THE MODE
F920    E606                    ANI     1<VDB%BE!1<VDB%IS
F922    CA F929                 JZ      ..NO7   ;NO BLINK OR DISPLAY INVERT
F925    78                      MOV     A,B
F926    F680                    ORI     80H
F928    47                      MOV     B,A
F929    78              ..NO7:  MOV     A,B     ;WRITE THE CHARACTER
F92A    D3E1                    OUT     VDBDAT
F92C    3EE0                    MVI     A,VDBXCR
F92E    D3E0                    OUT     VDBCTL  ;TEST X POS
F930    DBE0                    IN      VDBCTL
F932    B7                      ORA     A
F933    C0                      RNZ             ;NOPE
F934    3EE1                    MVI     A,VDBYCR
F936    D3E0                    OUT     VDBCTL  ;TEST Y POS
F938    DBE0                    IN      VDBCTL
F93A    B7                      ORA     A
F93B    C0                      RNZ             ;NOPE
F93C    CD F9E4         SCROL:  CALL    GTMD    ;GET CURRENT MODE
F93F    F5                      PUSH    PSW             ;FOR LATER
F940    F610                    ORI     1<VDB%DC        ;KILL CURSOR
F942    D3E0                    OUT     VDBCTL          ;INHIBIT
F944    E5                      PUSH    H       ;SAVE HL
F945    16C1                    MVI     D,VDBYCF+1
F947    0C                      INR     C       ;C=VDBDAT
```

```
F948    AF                      XRA     A       ;X=0
F949    D3E0                    OUT     VDBCTL
F94B    7A                      MOV     A,D     ;GET CURRENT Y
F94C    D3E0            ..SC1:  OUT     VDBCTL  ;SET Y POS
F94E    0650                    MVI     B,80    ;GET THE 80 CHARACTERS AT Y
F950    DBE1            ..AGN1: IN      VDBDAT
F952    77                      MOV     M,A
F953    23                      INX     H
F954    05                      DCR     B
F955    C2 F950                 JNZ     ..AGN1
F958    E1                      POP     H       ;RESET BUFF
F959    E5                      PUSH    H       ;SAVE AGAIN
F95A    7A                      MOV     A,D     ;Y=Y-1
F95B    3D                      DCR     A
F95C    D3E0                    OUT     VDBCTL
F95E    0650                    MVI     B,80
F960    7E              ..AGN2: MOV     A,M
F961    D3E1                    OUT     VDBDAT
F963    23                      INX     H
F964    05                      DCR     B
F965    C2 F960                 JNZ     ..AGN2
F968    E1                      POP     H
F969    E5                      PUSH    H
F96A    14                      INR     D
F96B    7A                      MOV     A,D
F96C    FED9                    CPI     25!VDBYCF
F96E    DA F94C                 JC      ..SC1
F971    E1                      POP     H
F972    CD F9ED                 CALL    CLIN    ;CLEAR LAST LINE
F975    3ED8                    MVI     A,24!VDBYCF
F977    D3E0                    OUT     VDBCTL
F979    F1                      POP     PSW
F97A    D3E0                    OUT     VDBCTL  ;ENABLE DISPLAY
F97C    C9                      RET
                        ;
F97D    FE0D            CNTL:   CPI     CR              ;IS IT CR?
F97F    CA F9D4                 JZ      ..CR            ;GO DO IT
F982    FE0A                    CPI     LF              ;IS IT LF?
F984    CA F9AE                 JZ      ..LF            ;GO DO IT
F987    FE08                    CPI     BS              ;IS IT BS?
F989    CA F9D8                 JZ      ..BS            ;GO DO IT
F98C    FE0C                    CPI     FF              ;IS IT FF?
F98E    CA F9BF                 JZ      ..FF            ;GO DO IT
F991    D611                    SUI     DC1             ;IS IT DC1?
F993    D8                      RC                      ;MISC. CNTL
F994    FE04                    CPI     4               ;IS IT DC4?
F996    D0                      RNC                     ;MISC. CONTROL
F997    0601                    MVI     B,1     ;THE SHIFT BIT
F999    3C                      INR     A
F99A    3D              ..SHFT: DCR     A
F99B    CA F9A7                 JZ      ..OK
F99E    F5                      PUSH    PSW
F99F    78                      MOV     A,B
F9A0    B7                      ORA     A       ;CLR CARRY
F9A1    17                      RAL
```

```
F9A2    47                      MOV     B,A
F9A3    F1                      POP     PSW
F9A4    C3 F99A                 JMP     ..SHFT  ;SET B UP
F9A7    CD F9E4         ..OK:   CALL    GTMD    ;GET CURRENT MODE
F9AA    A8                      XRA     B       ;COMPLIMENT SELECTED BIT
F9AB    D3E0                    OUT     VDBCTL  ;SET NEW MODE
F9AD    C9                      RET
                        ;
F9AE    3EE1            ..LF:   MVI     A,VDBYCR        ;ENABLE TO READ Y POS
F9B0    D3E0                    OUT     VDBCTL
F9B2    DBE0                    IN      VDBCTL  ;READ Y
F9B4    3C                      INR     A
F9B5    FE19                    CPI     25      ;TIME TO SCROLL?
F9B7    D2 F93C                 JNC     SCROL
F9BA    F6C0                    ORI     VDBYCF  ;SET Y
F9BC    D3E0                    OUT     VDBCTL
F9BE    C9                      RET
                        ;
F9BF    3EC0            ..FF:   MVI     A,VDBYCF
F9C1    D3E0                    OUT     VDBCTL          ;RESET X&Y
F9C3    AF                      XRA     A               ;
F9C4    D3E0                    OUT     VDBCTL          ;
F9C6    CD F9ED         ..FF1:  CALL    CLIN            ;CLEAR THE LINE
F9C9    3EE1                    MVI     A,VDBYCR        ;SET TO READ Y
F9CB    D3E0                    OUT     VDBCTL
F9CD    DBE0                    IN      VDBCTL          ;INPUT Y CURSOR
F9CF    B7                      ORA     A
F9D0    C2 F9C6                 JNZ     ..FF1           ;NOT DONE, TRY AGAIN
F9D3    C9                      RET                     ;DONE
                        ;
F9D4    AF              ..CR:   XRA     A       ;SET X=0
F9D5    D3E0                    OUT     VDBCTL
F9D7    C9                      RET
                        ;
F9D8    3EE0            ..BS:   MVI     A,VDBXCR
F9DA    D3E0                    OUT     VDBCTL
F9DC    DBE0                    IN      VDBCTL
F9DE    B7                      ORA     A
F9DF    C8                      RZ              ;AT LEFT MARGIN
F9E0    3D                      DCR     A
F9E1    D3E0                    OUT     VDBCTL  ;X=X-1
F9E3    C9                      RET
                        ;
F9E4    3EE2            GTMD:   MVI     A,VDBMRR        ;SET TO READ MODE
F9E6    D3E0                    OUT     VDBCTL          ;SET IT
F9E8    DBE0                    IN      VDBCTL          ;GET CURRENT MODE
F9EA    F680                    ORI     VDBMRF          ;SET TO WRITE MODE
F9EC    C9                      RET
                        ;
F9ED    3EE0            CLIN:   MVI     A,VDBXCR
F9EF    D3E0                    OUT     VDBCTL
F9F1    3E20            ..CL1:  MVI     A,' '
F9F3    D3E1                    OUT     VDBDAT
F9F5    DBE0                    IN      VDBCTL
F9F7    B7                      ORA     A
```

```
F9F8  C2 F9F1          JNZ     ..CL1
F9FB  C9               RET
                       ;
                       ; INITIALIZATION CODE. DRIVER MUST BE
                       ; IN RAM FOR THIS TO BE EFFECTIVE.
                       ;
0080                   MDBIT   = 80H   ;THIS CAN BE CHANGED TO ALLOW
                                       ;A USER DEFINED INITIALIZATION.
                                       ;
                                       ; 80H   = NORMAL MODE, BLINKING CURSOR
                                       ; 90H   = NORMAL MODE, NO CURSOR
                                       ; 88H   = REVERSE VIDEO, BLINKING CUR.
                                       ; 98H   = REVERSE VIDEO, NO CURSOR
                       ;
F9FC  3E80             INIT:   MVI     A,MDBIT ;WRITE TO MODE REG.
F9FE  D3E0                     OUT     VDBCTL
FA00  3E0C                     MVI     A,FF    ;FORM FEED
FA02  CD F97D                  CALL    CNTL    ;FAKE IT
FA05  21 F90B                  LXI     H,VDB
FA08  22 F904                  SHLD    MAIN+1
FA0B  E9                       PCHL            ;DO IT
                       ;
                       ;
                       ;
                       ;
                       ;       +++ VDB PARALLEL KEYBOARD INTERFACE +++
                       ;
                       ;       NOTE: Assumes jumpers from  'A' TO 'B'
                       ;             on VDB board. (See MANUAL)
                       ;
                       ;
FA0C  DBE2             KBIN:   IN      VDBK.S          ;READ STATUS
FA0E  E680                     ANI     80H             ;TEST FOR KEYPRESS
FA10  CA FA0C                  JZ      KBIN            ;NO, KEEP LOOKING
FA13  DBE3                     IN      VDBK.D          ;OK, READ IT
FA15  C9                       RET                     ;DONE
                       ;
FA16  DBE2             KBSTS:  IN      VDBK.S          ;READ STATUS
FA18  E680                     ANI     80H             ;TEST BIT 7
FA1A  3E00                     MVI     A,0             ;SET-UP A, NOT FLAGS
FA1C  C8                       RZ                      ;RETURN FALSE
FA1D  2F                       CMA
FA1E  C9                       RET                     ;RETURN TRUE
                       ;
                       ;
FA1F                   END.    == .
                       ;
                       .END
```

+++++ SYMBOL TABLE +++++

BEL	0007	BS	0008	BUFF	FC00	BUF.	FC00
CLIN	F9ED	CNTL	F97D	CR	000D	DC1	0011
DEL	007F	END.	FA1F	FF	000C	GTMD	F9E4
HT	0009	INIT	F9FC	KBIN	FA0C	KBSTS	FA16
LF	000A	LOC	F900	MAIN	F903	MDBIT	0080
SCROL	F93C	VDB	F90B	VDBCO	F900	VDBCTL	00E0
VDBDAT	00E1	VDBGMK	0060	VDBK.D	00E3	VDBK.S	00E2
VDBMRF	0080	VDBMRR	00E2	VDBRES	00E3	VDBXCR	00E0
VDBYCF	00C0	VDBYCR	00E1	VDB%AP	0000	VDB%BE	0001
VDB%DC	0004	VDB%DD	0005	VDB%ID	0003	VDB%IS	0002
ZAPPLE	F000						

```
                        ;
                                .REMARK "

                        THIS DRIVER IS DESIGNED TO INTERFACE THE
                        TDL ZAPPLE MONITOR WITH THE TDL VIDEO
                        DISPLAY BOARD (VDB). IT PROVIDES BOTH
                        A CHARACTER ORIENTED (TELETYPE-LIKE)
                        INTERFACE AND A X/Y PLOT INTERFACE.

                        Variable parameters are obtained at assembly
                        time. The RAM version buffer may be anywhere
                        in memory, and requires 80 decimal bytes.
                        The RAM program itself may be put in ROM,
                        but the self initializing feature must be
                        disabled.
                        "
                        ;
                        ;
                        ;
0001                    REL     =\      "RELOCATABLE (0-NO; 1-YES)"
                                .IFE    REL,[
                                .PABS
                        LOC     =\      "ENTER STARTING ADDRESS"
                                .LOC    LOC
                        ]
                        ;
                        ;
                        .PHEX
                        ;
                        ; COPYRIGHT 1977 By Technical Design Labs, Inc.
                        ; 1101 State Rd. Research Park, Bldg. H
                        ; Princeton, New Jersey  08540
                        ;
```

```
                        ;
                        ;       +++ VDB PORT AND MASK DEFINITIONS +++
                        ;
00E0                    VDBCTL  =\      "ENTER VDB STATUSPORT ADDRESS"
00E1                    VDBDAT  ==      VDBCTL+1        ;DATA PORT ADDRESS
00E2                    VDBK.S  ==      VDBCTL+2        ;VDB KEYBOARD STATUS
00E3                    VDBK.D  ==      VDBCTL+3        ;KEYBOARD DATA
0080                    VDBMRF  ==      10000000B       ;MODE REGISTER FLAG
00C0                    VDBYCF  ==      11000000B       ;Y CURSOR FLAG
00E0                    VDBXCR  ==      11100000B       ;X CURSOR READ
00E1                    VDBYCR  ==      11100001B       ;Y CURSOR READ
00E2                    VDBMRR  ==      11100010B       ;MODE REGISTER READ
00E3                    VDBRES  ==      11100011B       ;RESET VDB
0060                    VDBGMK  ==      01100000B       ;GRAPHIC MASK
                        ;
                        ;       +++ MODE BIT DEFINITIONS +++
                        ;
0000                    VDB%AP  ==      0               ;ALTERNATE PAGE
0001                    VDB%BE  ==      1               ;BLINK ENABLE
0002                    VDB%IS  ==      2               ;INVERT SYMBOL
0003                    VDB%ID  ==      3               ;INVERT DISPLAY
0004                    VDB%DC  ==      4               ;DISABLE CURSOR
0005                    VDB%DD  ==      5               ;DISABLE DISPLAY
                        ;
                        ;       +++ ASCII CONTROL CHARACTERS +++
                        ;
0008                    BS      ==      08H             ;BACK SPACE
0009                    HT      ==      09H             ;TAB
000D                    CR      ==      0DH             ;CARRIAGE RETURN
000A                    LF      ==      0AH             ;LINE FEED
000C                    FF      ==      0CH             ;FORM FEED
0007                    BEL     ==      07H             ;BELL
0011                    DC1     ==      11H             ;DC CONTROLS
007F                    DEL     ==      7FH             ;DELETE
                        ;
```

ROM/Z80


```
                        ;
                        ;       +++ VDB TELETYPE SIMULATOR +++
                        ;
0000'  C5               VDBCO:  PUSH    B               ;SAVE REGISTERS
0001'  D5                       PUSH    D               ;
0002'  E5                       PUSH    H               ;
0003'  CD 000B'                 CALL    VDB
0006'  E1                       POP     H               ;RESTORE REGISTERS
0007'  D1                       POP     D               ;
0008'  C1                       POP     B               ;
0009'  79                       MOV     A,C             ;OUTPUT CHARACTER INTO

000A'  C9                       RET                     ;DONE
                        ;
                        ;       +++ ROM VDB OUTPUT DRIVER +++
                        ;
000B'  79               VDB:    MOV     A,C             ;GET OUTPUT CHARACTER
000C'  E67F                     ANI     7FH             ;GET RID OF PARITY BIT
000E'  C8                       RZ                      ;IGNORE NULLS
000F'  FE7F                     CPI     DEL             ;RUBOUT?
0011'  C8                       RZ                      ;IGNORE IT
0012'  01 C0E0                  LXI     B,VDBYCF<8!VDBCTL ;SET UP CONSTANTS
0015'  21 E0E1                  LXI     H,VDBXCR<8!VDBYCR ;CURSOR READ OPS
0018'  FE20                     CPI     ' '             ;IS IT A CONTROL CHARAC
                        ER?
001A'  386F                     JRC     ..CNTL
001C'  D3E1                     OUT     VDBDAT          ;OUTPUT THE CHARACTER
001E'  ED69                     OUTP    L               ;SET TO READ Y CURSOR
0020'  ED58                     INP     E               ;READ IT
0022'  C0                       RNZ                     ;NOT AT FIRST LINE
0023'  ED61                     OUTP    H               ;SET TO READ X CURSOR
0025'  ED50                     INP     D               ;READ IT
0027'  C0                       RNZ                     ;NOT FIRST CHARACTER
0028'  3EE2             ..SCRL: MVI     A,VDBMRR        ;SET TO READ MODE
002A'  D3E0                     OUT     VDBCTL          ;SET IT
002C'  DBE0                     IN      VDBCTL          ;GET CURRENT MODE
002E'  F680                     ORI     VDBMRF          ;SET TO WRITE MODE
0030'  F5                       PUSH    PSW             ;FOR LATER
0031'  F610                     ORI     1<VDB%DC        ;KILL CURSOR
0033'  D3E0                     OUT     VDBCTL          ;INHIBIT
0035'  ED61                     OUTP    H               ;SET TO READ COLUMN CUR
                        OR      INR     B
0037'  58               ..SCL1: MOV     E,B             ;Y1 <- 0 LINE
0038'  04                       INR     B               ;Y2 <- Y1+1
0039'  ED51             ..SCL2: OUTP    D               ;SET X CURSOR
003B'  ED41                     OUTP    B               ;SET Y<-READ LINE (Y2)
003D'  DBE1                     IN      VDBDAT          ;GET CHARACTER THERE
003F'  ED51                     OUTP    D               ;RESET X CURSOR
0041'  ED59                     OUTP    E               ;SET Y<-WRITE LINE (Y1)
0043'  D3E1                     OUT     VDBDAT          ;OUTPUT THE CHARACTER
0045'  ED50                     INP     D               ;READ X CURSOR
0047'  20F0                     JRNZ    ..SCL2          ;NOT DONE
0049'  78                       MOV     A,B             ;GET Y2 (READ) LINE
004A'  D6D8                     SUI     24!VDBYCF       ;ARE WE DONE?
004C'  20E9                     JRNZ    ..SCL1          ;NO, ONWARD
```

```
004E'  CD 00C6'            CALL    .CLIN      ;CLEAR THE LINE
0051'  ED41                OUTP    B          ;RESET Y TO LINE 24
0053'  182E                JMPR    .LF1       ;GO RETURN
0055'  ED61          BS:   OUTP    H          ;SET TO READ X CURSOR
0057'  ED50                INP     D          ;GET IT
0059'  2804                JRZ     .BS2       ;AT START OF LINE
005B'  15                  DCR     D          ;BACK UP
005C'  ED51          .BS3: OUTP    D          ;OUTPUT NEW CURSOR
005E'  C9                  RET                ;DONE
005F'  ED69          .BS2: OUTP    L          ;SET TO READ Y CURSOR
0061'  ED58                INP     E          ;READ Y
0063'  C8                  RZ                 ;AT UPPER LEFT
0064'  1D                  DCR     E          ;ONE LESS
0065'  7B                  MOV     A,E        ;GET IT
0066'  B0                  ORA     B          ;PUT IN Y FLAG
0067'  D3E0                OUT     VDBCTL     ;SET Y THERE
0069'  164F                MVI     D,79       ;AT END OF LINE
006B'  18EF                JMPR    .BS3       ;OUTPUT NEW X
006D'  AF            ..CR: XRA     A          ;SET X TO ZERO
006E'  D3E0                OUT     VDBCTL     ;OUTPUT IT
0070'  C9                  RET                ;DONE
0071'  ED69          ..LF: OUTP    L          ;SET TO READ Y
0073'  DBE0                IN      VDBCTL     ;READ Y CURSOR
0075'  FE18                CPI     24         ;AT LAST LINE NOW?
0077'  200D                JRNZ    ..LF2      ;NO, DONT SCROLL
0079'  ED61                OUTP    H          ;SET TO READ X
007B'  DBE0                IN      VDBCTL     ;GET X CURSOR
007D'  F5                  PUSH    PSW        ;SAVE IT
007E'  1600                MVI     D,0        ;INIT COLUMN FOR SCROL
0080'  CD 0028'            CALL    .SCRL      ;SCROLL THE SCREEN
0083'  F1            .LF1: POP     PSW        ;GET X BACK
0084'  1802                JMPR    .LF3       ;GO SET X
0086'  3C            .LF2: INR     A          ;NEXT LINE
0087'  B0                  ORA     B          ;SET Y CURSOR FLAG
0088'  D3E0          .LF3: OUT     VDBCTL     ;OUTPUT CURSOR
008A'  C9                  RET                ;DONE
008B'  FE0D          .CNTL: CPI    CR         ;IS IT CR?
008D'  28DE                JRZ     .CR        ;GO DO IT
008F'  FE0A                CPI     LF         ;IS IT LF?
0091'  28DE                JRZ     .LF        ;GO DO IT
0093'  FE08                CPI     BS         ;IS IT BS?
0095'  28BE                JRZ     .BS        ;GO DO IT
0097'  FE0C                CPI     FF         ;IS IT FF?
0099'  281C                JRZ     .FF        ;GO DO IT
009B'  D611                SUI     DC1        ;IS IT DC1?
009D'  D8                  RC                 ;MISC. CNTL
009E'  FE04                CPI     4          ;IS IT DC4?
00A0'  D0                  RNC                ;MISC. CONTROL
00A1'  0601                MVI     B,1        ;THE SHIFT BIT
00A3'  3C                  INR     A
00A4'  3D            .SHFT: DCR    A
00A5'  2804                JRZ     .OK
00A7'  CB20                SLAR    B
00A9'  18F9                JMPR    .SHFT      ;SET B UP
00AB'  3EE2          .OK:  MVI     A,VDBMRR   ;SET TO READ MODE
```

```
00AD'   D3E0                    OUT     VDBCTL          ;SET IT
00AF'   DBE0                    IN      VDBCTL          ;GET CURRENT MODE
00B1'   A8                      XRA     B               ;COMPLIMENT SELECTED BI

00B2'   F680                    ORI     VDBMRF          ;SET TO WRITE MODE
00B4'   D3E0                    OUT     VDBCTL          ;SET NEW MODE
00B6'   C9                      RET
                        ;
00B7'   ED41            ..FF:   OUTP    B               ;RESET ROW TO ZERO
00B9'   AF                      XRA     A               ;
00BA'   D3E0                    OUT     VDBCTL          ;
00BC'   CD 00C6'        ..FF1:  CALL    ..CLIN          ;CLEAR THE LINE
00BF'   ED69                    OUTP    L               ;SET TO READ Y
00C1'   ED58                    INP     E               ;INPUT Y CURSOR
00C3'   20F7                    JRNZ    ..FF1           ;NOT DONE, TRY AGAIN
00C5'   C9                      RET                     ;DONE
00C6'   ED61            ..CLIN: OUTP    H               ;SET TO READ X CURSOR
00C8'   D3E1            ..CLNP: OUT     VDBDAT          ;OUTPUT SPACE
00CA'   ED50                    INP     D               ;GET X CURSOR
00CC'   20FA                    JRNZ    ..CLNP          ;NOT DONE
00CE'   C9                      RET                     ;DONE
                        ;
                        ;       +++ VDB PARALLEL KEYBOARD INTERFACE +++
                        ;
                        ;       NOTE: Assumes jumpers from  'A' TO 'B'
                        ;             on VDB board. (See MANUAL)
                        ;
                        ;
00CF'   DBE2            KBIN:   IN      VDBK.S          ;READ STATUS
00D1'   E680                    ANI     80H             ;TEST FOR KEYPRESS
00D3'   28FA                    JRZ     KBIN            ;NO, KEEP LOOKING
00D5'   DBE3                    IN      VDBK.D          ;OK, READ IT
00D7'   C9                      RET                     ;DONE
                        ;
00D8'   DBE2            KBSTS:  IN      VDBK.S          ;READ STATUS
00DA'   E680                    ANI     80H             ;TEST BIT 7
00DC'   3E00                    MVI     A,0             ;SET-UP A, NOT FLAGS
00DE'   C8                      RZ                      ;RETURN FALSE
00DF'   2F                      CMA
00E0'   C9                      RET                     ;RETURN TRUE
                        ;
                        ;
                        .END
```

+++++ SYMBOL TABLE +++++

BEL	0007	BS	0008	CR	000D	DC1	0011
DEL	007F	FF	000C	HT	0009	KBIN	00CF'
KBSTS	00D8'	LF	000A	REL	0001	VDB	000B'
VDBCO	0000'	VDBCTL	00E0	VDBDAT	00E1	VDBGMK	0060
VDBK.D	00E3	VDBK.S	00E2	VDBMRF	0080	VDBMRR	00E2
VDBRES	00E3	VDBXCR	00E0	VDBYCF	00C0	VDBYCR	00E1
VDB%AP	0000	VDB%BE	0001	VDB%DC	0004	VDB%DD	0005
VDB%ID	0003	VDB%IS	0002				

```
                        ;
                                .REMARK "

                        THIS DRIVER IS DESIGNED TO INTERFACE THE
                        TDL ZAPPLE MONITOR WITH THE TDL VIDEO
                        DISPLAY BOARD (VDB). IT PROVIDES BOTH
                        A CHARACTER ORIENTED (TELETYPE-LIKE)
                        INTERFACE AND A X/Y PLOT INTERFACE.

                        Variable parameters are obtained at assembly
                        time. The RAM version buffer may be anywhere
                        in memory, and requires 80 decimal bytes.
                        The RAM program itself may be put in ROM,
                        but the self initializing feature must be
                        disabled.
                        "
                        ;
                        ;
                        ;       **RAM** VERSION
                        ;
FC00                    BUF.    =\      "BUFFER LOCATION?"
                        ;
                        ;
                                .IFN    BUF.,[
FC00                    BUFF    =       BUF.
                        ]
                        [
                        BUFF    =       END.
                        ]
                        ;
                        .PHEX
                        .XLINK
                        ;
                        ;
0001                    REL     =\      "RELOCATABLE (0-NO; 1-YES)"
                                .IFE    REL,[
                                .PABS
                        LOC     =\      "ENTER STARTING ADDRESS"
                                .LOC    LOC
                        ]
                        ;
                        ; COPYRIGHT 1977 By Technical Design Labs, Inc.
                        ; 1101 State Rd. Research Park, Bldg. H
                        ; Princeton, New Jersey  08540
                        ;
```

```
                        ;
                        ;
                        ;       THIS SECTION SETS THE LOAD MODULE UP
                        ;       SO THAT IT WILL AUTOMATICALLY PLUG
                        ;       THE "USER" VECTORS TO USE THE VDB.
                        ;
0000'                   RZERO   == .            ;A RELOCATABLE ZERO
                        ;
                        ;       +++ ZAPPLE MONITOR EQUATES +++
                        ;
F000                    ZAPPLE  ==      0F000H          ;ZAPPLE MONITOR LOCATI

                        ;
                        ;
F800                    .LOC    ZAPPLE+800H             ;INSERT JMPS FOR USER
                        NSOLE
                        ;
F800    C3 00DA'        JMP     KBIN    ;USER CONSOLE INPUT
F803    C3 0000'        JMP     VDBCO   ;USER CONSOLE OUTPUT
                        ;
F815                    .LOC    ZAPPLE+815H
                        ;
F815    C3 0000'        JMP     VDBCO   ;USER LIST DEVICE
F818    C3 00E3'        JMP     KBSTS   ;USER LIST DEVICE
                        ;
0000'                   .LOC    RZERO
                        ;
                        ;       +++ ASCII CONTROL CHARACTERS +++
                        ;
0008                    BS      ==      08H             ;BACK SPACE
0009                    HT      ==      09H             ;TAB
000D                    CR      ==      0DH             ;CARRIAGE RETURN
000A                    LF      ==      0AH             ;LINE FEED
000C                    FF      ==      0CH             ;FORM FEED
0007                    BEL     ==      07H             ;BELL
0011                    DC1     ==      11H             ;DC CONTROLS
007F                    DEL     ==      7FH             ;DELETE
                        ;
```

```
                        ;
                        ;
                        ;       +++ VDB PORT AND MASK DEFINITIONS +++
                        ;
00E0                    VDBCTL  =\      "ENTER VDB STATUSPORT ADDRESS"
00E1                    VDBDAT  ==      VDBCTL+1        ;DATA PORT ADDRESS
00E2                    VDBK.S  ==      VDBCTL+2        ;VDB KEYBOARD STATUS
00E3                    VDBK.D  ==      VDBCTL+3        ;KEYBOARD DATA
0080                    VDBMRF  ==      10000000B       ;MODE REGISTER FLAG
00C0                    VDBYCF  ==      11000000B       ;Y CURSOR FLAG
00E0                    VDBXCR  ==      11100000B       ;X CURSOR READ
00E1                    VDBYCR  ==      11100001B       ;Y CURSOR READ
00E2                    VDBMRR  ==      11100010B       ;MODE REGISTER READ
00E3                    VDBRES  ==      11100011B       ;RESET VDB
0060                    VDBGMK  ==      01100000B       ;GRAPHIC MASK
                        ;
                        ;       +++ MODE BIT DEFINITIONS +++
                        ;
0000                    VDB%AP  ==      0               ;ALTERNATE PAGE
0001                    VDB%BE  ==      1               ;BLINK ENABLE
0002                    VDB%IS  ==      2               ;INVERT SYMBOL
0003                    VDB%ID  ==      3               ;INVERT DISPLAY
0004                    VDB%DC  ==      4               ;DISABLE CURSOR
0005                    VDB%DD  ==      5               ;DISABLE DISPLAY
                        ;
                        ;       +++ VDB TELETYPE SIMULATOR +++
                        ;
0000'   C5              VDBCO:  PUSH    B               ;SAVE REGISTERS
0001'   D5                      PUSH    D               ;
0002'   E5                      PUSH    H               ;
0003    CD 00EC'        MAIN:   CALL    INIT            ;CALL DRIVER
0006'   E1                      POP     H               ;RESTORE REGISTERS
0007'   D1                      POP     D               ;
0008'   C1                      POP     B               ;
0009'   79                      MOV     A,C             ;OUTPUT CHARACTER INTO

000A'   C9                      RET                     ;DONE
                        ;
                        ;       +++ RAM VDB DRIVER +++
                        ;
000B'   79              VDB:    MOV     A,C             ;GET OUTPUT CHARACTER
000C'   E67F                    ANI     7FH             ;GET RID OF PARITY BIT
000E'   C8                      RZ                      ;IGNORE NULLS
000F'   FE7F                    CPI     DEL             ;RUBOUT?
0011'   C8                      RZ                      ;IGNORE IT
0012'   0EE0                    MVI     C,VDBCTL        ;SET C UP
0014'   21 FC00                LXI     H,BUFF          ;POINT TO BUFFER
0017'   FE20                    CPI     " "     ;CONTROL CHARACTER?
0019'   384E                    JRC     CNTL    ;YES
001B'   47                      MOV     B,A     ;SAVE PRINTING CHAR.
001C'   CD 00C3'               CALL    GTMD    ;GET THE MODE
001F'   E606                    ANI     1<VDB%BE!1<VDB%IS
0021'   2802                    JRZ     ..NO7   ;NO BLINK OR DISPLAY INVERT
0023'   CBF8                    SET     7,B
0025'   78              ..NO7:  MOV     A,B     ;WRITE THE CHARACTER
```

```
0026'   D3E1                OUT     VDBDAT
0028'   3EE0                MVI     A,VDBXCR
002A'   D3E0                OUT     VDBCTL  ;TEST X POS
002C'   ED40                INP     B       ;SEE IF WE JUST WENT ZERO
002E'   C0                  RNZ             ;NOPE
002F'   3C                  INR     A       ;A=VDBYCR
0030'   D3E0                OUT     VDBCTL  ;TEST Y POS
0032'   ED40                INP     B       ;DID WE GO ZERO?
0034'   C0                  RNZ             ;NOPE
0035'   CD 00C3'     SCROL:  CALL    GTMD    ;GET CURRENT MODE
0038'   F5                  PUSH    PSW             ;FOR LATER
0039'   F610                ORI     1<VDB%DC        ;KILL CURSOR
003B'   D3E0                OUT     VDBCTL          ;INHIBIT
003D'   E5                  PUSH    H       ;SAVE HL
003E'   16C1                MVI     D,VDBYCF+1
0040'   0C                  INR     C       ;C=VDBDAT
0041'   AF                  XRA     A       ;X=0
0042'   D3E0                OUT     VDBCTL
0044'   7A                  MOV     A,D     ;GET CURRENT Y
0045'   D3E0          ..SC1:OUT     VDBCTL  ;SET Y POS
0047'   0650                MVI     B,80    ;GET THE 80 CHARACTERS AT Y
0049'   EDB2                INIR
004B'   E1                  POP     H       ;RESET BUFF
004C'   E5                  PUSH    H       ;SAVE AGAIN
004D'   7A                  MOV     A,D     ;Y=Y-1
004E'   3D                  DCR     A
004F'   D3E0                OUT     VDBCTL
0051'   0650                MVI     B,80
0053'   EDB3                OUTIR
0055'   E1                  POP     H
0056'   E5                  PUSH    H
0057'   14                  INR     D
0058'   7A                  MOV     A,D
0059'   FED9                CPI     25!VDBYCF
005B'   38E8                JRC     ..SC1
005D'   E1                  POP     H
005E'   CD 00CC'            CALL    CLIN    ;CLEAR LAST LINE
0061'   3ED8                MVI     A,24!VDBYCF
0063'   D3E0                OUT     VDBCTL
0065'   F1                  POP     PSW
0066'   D3E0                OUT     VDBCTL  ;ENABLE DISPLAY
0068'   C9                  RET
                    ;
0069'   FE0D        CNTL:   CPI     CR              ;IS IT CR?
006B'   2847                JRZ     ..CR            ;GO DO IT
006D'   FE0A                CPI     LF              ;IS IT LF?
006F'   281F                JRZ     ..LF            ;GO DO IT
0071'   FE08                CPI     BS              ;IS IT BS?
0073'   2843                JRZ     ..BS            ;GO DO IT
0075'   FE0C                CPI     FF              ;IS IT FF?
0077'   2827                JRZ     ..FF            ;GO DO IT
0079'   D611                SUI     DC1             ;IS IT DC1?
007B'   D8                  RC                      ;MISC. CNTL
007C'   FE04                CPI     4               ;IS IT DC4?
007E'   D0                  RNC                     ;MISC. CONTROL
```

```
007F'   0601                    MVI     B,1     ;THE SHIFT BIT
0081'   3C                      INR     A
0082'   3D            ..SHFT:   DCR     A
0083'   2804                    JRZ     ..OK
0085'   CB20                    SLAR    B
0087'   18F9                    JMPR    ..SHFT  ;SET B UP
0089'   CD 00C3'      ..OK:     CALL    GTMD    ;GET CURRENT MODE
008C'   A8                      XRA     B       ;COMPLIMENT SELECTED BIT
008D'   D3E0                    OUT     VDBCTL  ;SET NEW MODE
008F'   C9                      RET
                        ;
0090'   3EE1            ..LF:   MVI     A,VDBYCR        ;ENABLE TO READ Y POS
0092'   D3E0                    OUT     VDBCTL
0094'   DBE0                    IN      VDBCTL  ;READ Y
0096'   3C                      INR     A
0097'   FE19                    CPI     25      ;TIME TO SCROLL?
0099'   309A                    JRNC    SCROL
009B'   F6C0                    ORI     VDBYCF  ;SET Y
009D'   D3E0                    OUT     VDBCTL
009F'   C9                      RET
                        ;
00A0'   3EC0            ..FF:   MVI     A,VDBYCF
00A2'   D3E0                    OUT     VDBCTL          ;RESET X&Y
00A4'   AF                      XRA     A               ;
00A5'   D3E0                    OUT     VDBCTL          ;
00A7'   CD 00CC'        ..FF1:  CALL    CLIN            ;CLEAR THE LINE
00AA'   3EE1                    MVI     A,VDBYCR        ;SET TO READ Y
00AC'   D3E0                    OUT     VDBCTL
00AE'   DBE0                    IN      VDBCTL          ;INPUT Y CURSOR
00B0'   B7                      ORA     A
00B1'   20F4                    JRNZ    ..FF1           ;NOT DONE, TRY AGAIN
00B3'   C9                      RET                     ;DONE
                        ;
00B4'   AF              ..CR:   XRA     A       ;SET X=0
00B5'   D3E0                    OUT     VDBCTL
00B7'   C9                      RET
                        ;
00B8'   3EE0            ..BS:   MVI     A,VDBXCR
00BA'   D3E0                    OUT     VDBCTL
00BC'   ED58                    INP     E
00BE'   C8                      RZ              ;AT LEFT MARGIN
00BF'   1D                      DCR     E
00C0'   ED59                    OUTP    E       ;X=X-1
00C2'   C9                      RET
                        ;
00C3'   3EE2            GTMD:   MVI     A,VDBMRR        ;SET TO READ MODE
00C5'   D3E0                    OUT     VDBCTL          ;SET IT
00C7'   DBE0                    IN      VDBCTL          ;GET CURRENT MODE
00C9'   F680                    ORI     VDBMRF          ;SET TO WRITE MODE
00CB'   C9                      RET
                        ;
00CC'   3EE0            CLIN:   MVI     A,VDBXCR
00CE'   D3E0                    OUT     VDBCTL
00D0'   3E20            ..CL1:  MVI     A,' '
00D2'   D3E1                    OUT     VDBDAT
```

```
00D4'   DBE0                    IN      VDBCTL
00D6'   B7                      ORA     A
00D7'   20F7                    JRNZ    ..CL1
00D9'   C9                      RET
                        ;
                        ;
                        ;       +++ VDB PARALLEL KEYBOARD INTERFACE +++
                        ;
                        ;       NOTE: Assumes jumpers from  'A' TO 'B'
                        ;             on VDB board. (See MANUAL)
                        ;
                        ;
00DA'   DBE2            KBIN:   IN      VDBK.S          ;READ STATUS
00DC'   E680                    ANI     80H             ;TEST FOR KEYPRESS
00DE'   28FA                    JRZ     KBIN            ;NO, KEEP LOOKING
00E0'   DBE3                    IN      VDBK.D          ;OK, READ IT
00E2'   C9                      RET                     ;DONE
                        ;
00E3'   DBE2            KBSTS:  IN      VDBK.S          ;READ STATUS
00E5'   E680                    ANI     80H             ;TEST BIT 7
00E7'   3E00                    MVI     A,0             ;SET-UP A, NOT FLAGS
00E9'   C8                      RZ                      ;RETURN FALSE
00EA'   2F                      CMA
00EB'   C9                      RET                     ;RETURN TRUE
                        ;
                        ;
                        ; INITIALIZATION CODE. DRIVER MUST BE
                        ; IN RAM FOR THIS TO BE EFFECTIVE.
                        ;
0080                    MDBIT   = 80H   ;THIS CAN BE CHANGED TO ALLOW
                                        ;A USER DEFINED INITIALIZATION.
                                        ;
                                        ; 80H   = NORMAL MODE, BLINKING CURSOR
                                        ; 90H   = NORMAL MODE, NO CURSOR
                                        ; 88H   = REVERSE VIDEO, BLINKING CUR
                                        ; 98H   = REVERSE VIDEO, NO CURSOR
                        ;
00EC'   3E80            INIT:   MVI     A,MDBIT ;WRITE TO MODE REG.
00EE'   D3E0                    OUT     VDBCTL
00F0'   3E0C                    MVI     A,FF    ;FORM FEED
00F2'   CD 0069'                CALL    CNTL    ;FAKE IT
00F5'   21 000B'                LXI     H,VDB
00F8'   22 0004'                SHLD    MAIN+1
00FB'   E9                      PCHL            ;DO IT
                        ;
                        ;
                        ;
                        ;
```

```
                        ;
                        ;       BOOTSTRAP SUPPORT
                        ;
                        ;
00FC'   3EE3            BOOT:   MVI     A,0E3H          ;RESET VDB
00FE'   D3E0                    OUT     VDBCTL
0100'   DB76                    IN      076H            ;CURRENT I/O CONFIG.
0102'   F603                    ORI     3               ;SET TO USER CONSOLE
0104'   D376                    OUT     076H
0106'   C3 F057                JMP     0F057H          ;SIGN-ON ZAPPLE
                        ;
                        ;
00FC'                   .END    BOOT
```

+++++ SYMBOL TABLE +++++

BEL	0007	BOOT	00FC'	BS	0008	BUFF	FC00
BUF.	FC00	CLIN	00CC'	CNTL	0069'	CR	000D
DC1	0011	DEL	007F	FF	000C	GTMD	00C3'
HT	0009	INIT	00EC'	KBIN	00DA'	KBSTS	00E3'
LF	000A	MAIN	0003'	MDBIT	0080	REL	0001
RZERO	0000'	SCROL	0035'	VDB	000B'	VDBCO	0000'
VDBCTL	00E0	VDBDAT	00E1	VDBGMK	0060	VDBK.D	00E3
VDBK.S	00E2	VDBMRF	0080	VDBMRR	00E2	VDBRES	00E3
VDBXCR	00E0	VDBYCF	00C0	VDBYCR	00E1	VDB%AP	0000
VDB%BE	0001	VDB%DC	0004	VDB%DD	0005	VDB%ID	0003
VDB%IS	0002	ZAPPLE	F000				

```
                        ;
                        ; This routine expects to be "CALLed" from BASIC
                        ; with 3 parameters:
                        ;
                        ; CALL <ENTRY>,[P1],[P2],[P3]
                        ;
                        ; Where P1=r/w info (1=on, 0=off).
                        ;       P2='X' location (0 thru 159).
                        ;       P3='Y' location (0 thru 74).
                        ;
0000'   79              ENTRY:  MOV     A,C     ;Get number of parameters
0001'   FE03                    CPI     3       ;Should be 3
0003'   2802                    JRZ     ..OK
0005'   F9                      SPHL            ;ABORT
0006'   C9                      RET
                        ;
0007'   D1              ..OK:   POP     D       ;Get 'y' info
0008'   7B                      MOV     A,E     ;Save 'y' info
0009'   D1                      POP     D       ;Get 'x' info
000A'   53                      MOV     D,E     ;'x' in D
000B'   5F                      MOV     E,A     ;Restore 'y' info
000C'   C1                      POP     B       ;Get r/w info
000D'   79                      MOV     A,C
                        ;
                        ; Continue & return
                        ;
                        ;
                        ;       +++ VDB GRAPHICS DRIVER +++
                        ;
000E'   F5              VDBPLT: PUSH    PSW             ;SAVE ON/OFF INFO
000F'   7B                      MOV     A,E             ;GET Y COORDINATE
0010'   01 03D9                 LXI     B,3D9H          ;set for divide
0013'   90              ..DVLP: SUB     B               ;DIVIDE BY 3
0014'   0D                      DCR     C               ;  GET 24-QUOTIENT
0015'   30FC                    JRNC    ..DVLP          ;
0017'   80                      ADD     B               ;GET REMAINDER
0018'   87                      ADD     A               ;LEFT SHIFT 1
0019'   5F                      MOV     E,A             ;E<-RR*2
001A'   AF                      XRA     A               ;CLEAR A/CARRY
001B'   47                      MOV     B,A             ;CLEAR MSB OF OFFSET
001C'   CB3A                    SRLR    D               ;COLUMN<-X/2
001E'   8B                      ADC     E               ;A<-RR*2+CR
001F'   59                      MOV     E,C             ;E<-ROW
0020'   4F                      MOV     C,A             ;BC<-OFFSET
0021'   21 0068'                LXI     H,..MASK        ;MASK FOR THAT DOT
0024'   09                      DAD     B               ;POINT TO IT
0025'   66                      MOV     H,M             ;SAVE IT
0026'   F1                      POP     PSW             ;GET ON/OFF INFO
0027'   B7                      ORA     A               ;0=OFF, <>0=ON
0028'   2EFF                    MVI     L,-1
002A'   2001                    JRNZ    ..SK
002C'   2C                      INR     L               ;L=ON/OFF NOW
002D'   0EE0            ..SK:   MVI     C,0E0H          ;CONTROL PORT
002F'   ED51                    OUTP    D               ;X CURSOR
```

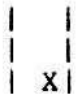
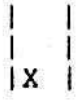
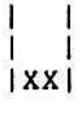
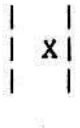
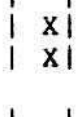
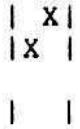
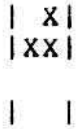
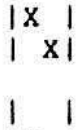
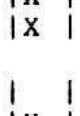
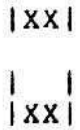
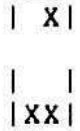
```
0031'   ED59                    OUTP    E               ;Y CURSOR
0033'   DBE1                    IN      0E1H            ;GET WHATS THERE
0035'   47                      MOV     B,A             ;SAVE CHARACTER
0036'   ED51                    OUTP    D               ;SET CURSOR BACK
0038'   ED59                    OUTP    E               ;
003A'   CB7C                    BIT     7,H             ;SPECIAL CASE
003C'   2813                    JRZ     ..OK2
003E'   E680                    ANI     80H             ;0=OFF, 1=ON
0040'   2807                    JRZ     ..OFF
0042'   CB45                    BIT     0,L             ;SEE WHAT WE WANT
0044'   78                      MOV     A,B             ;RESTORE CHARACTER
0045'   201E                    JRNZ    ..OK            ;ALREADY ON
0047'   1805                    JMPR    ..OK1
0049'   CB45            ..OFF:  BIT     0,L             ;SEE WHAT WE WANT
004B'   78                      MOV     A,B
004C'   2817                    JRZ     ..OK            ;ALREADY OFF
004E'   AC              ..OK1:  XRA     H               ;TURN IT OFF
004F'   1814                    JMPR    ..OK
0051'   7C              ..OK2:  MOV     A,H             ;GET MASK
0052'   57                      MOV     D,A             ;"ON" MASK
0053'   2F                      CMA
0054'   5F                      MOV     E,A             ;"OFF" MASK
0055'   CB78                    BIT     7,B             ;SEE IF SPECIAL
0057'   2803                    JRZ     ..OK3           ; NO
0059'   7D                      MOV     A,L             ; YES, INVERT SENSE
005A'   2F                      CMA
005B'   6F                      MOV     L,A
005C'   78              ..OK3:  MOV     A,B             ;GET CHARACTER
005D'   A3                      ANA     E               ;UNCONDITIONAL OFF
005E'   47                      MOV     B,A             ;SAVE AGAIN
005F'   CB45                    BIT     0,L             ;GET ON/OFF INFO
0061'   78                      MOV     A,B             ;GET THE CHARACTER
0062'   2801                    JRZ     ..OK            ;LEAVE IT OFF
0064'   B2                      ORA     D               ;TURN IT ON
0065'   D3E1            ..OK:   OUT     0E1H            ;OUTPUT CHARACTER
0067'   C9                      RET                     ;
                        ;
0068'   020108          ..MASK: .BYTE   00000010B,00000001B,00001000B
006B'   049F10                  .BYTE   00000100B,10011111B,00010000B
                        ;
                        ;
                        ;
                        .END
```

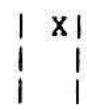
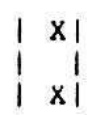
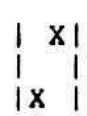
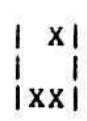
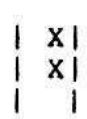
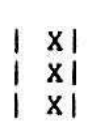
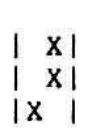
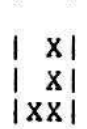
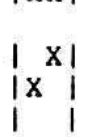
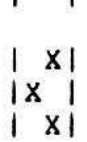
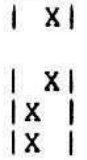
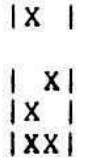
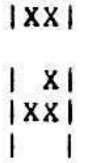
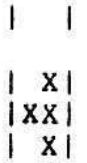
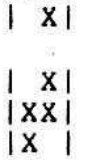
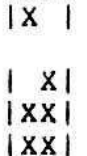
+++++ SYMBOL TABLE +++++

ENTRY 0000' VDBPLT 000E'

The following is a sample basic program, using the above driver, which was loaded at 80H. (R,80) It will demonstrate some possible methods of using the VDB's graphics capability.

```
10 E0=&E0:E1=&E1:DR=&80
20 LPRINT CHR$(12):OUT E0,&90
30 FOR N=0 TO 9:READ I(N):NEXT
40 LPRINT, "Average rainfall for Podunk & vicinity":LPRINT
50 FOR I=60 TO 0 STEP -3
60 LPRINT USING "###";I:NEXT
70 LPRINT "     ";
80 FOR I=0 TO 9
90 LPRINT USING "  197#";I;:NEXT
100 A$="INCHES/YEAR":X=140
110 FOR I=1 TO LEN(A$)
120 CALL DR,0,X,60-(3*I)
130 OUT E1,ASC(MID$(A$,I,1)):NEXT:X=6
140 FOR T=0 TO 9
150 V=I(T):X=X+12
160 FOR I=3 TO V+3
170 CALL DR,1,X,I:CALL DR,1,X-1,I:IF X>110 THEN CALL DR,0,X,I-1
180 CALL DR,1,X+1,I:NEXT
190 CALL DR,1,X,I-1:NEXT
200 DATA 55,43,32,59,48,22,12,42,45,38
```

NUMBER	BINARY	HEXADECIMAL	PATTERN
1	00000000B	00H	
2	00000001B	01H	
3	00000010B	02H	
4	00000011B	03H	
5	00000100B	04H	
6	00000101B	05H	
7	00000110B	06H	
8	00000111B	07H	
9	00001000B	08H	
10	00001001B	09H	
11	00001010B	0AH	
12	00001011B	0BH	
13	00001100B	0CH	
14	00001101B	0DH	
15	00001110B	0EH	
16	00001111B	0FH	

NUMBER	BINARY	HEXADECIMAL	PATTERN
17	00010000B	10H	
18	00010001B	11H	
19	00010010B	12H	
20	00010011B	13H	
21	00010100B	14H	
22	00010101B	15H	
23	00010110B	16H	
24	00010111B	17H	
25	00011000B	18H	
26	00011001B	19H	
27	00011010B	1AH	
28	00011011B	1BH	
29	00011100B	1CH	
30	00011101B	1DH	
31	00011110B	1EH	
32	00011111B	1FH	

NUMBER	BINARY	HEXADECIMAL	PATTERN
49	10001111B	8FH	|XX| | | | |
50	10001110B	8EH	|XX| | | | X|
51	10001101B	8DH	|XX| | | |X |
52	10001100B	8CH	|XX| | | |XX|
53	10001011B	8BH	|XX| | X| | |
54	10001010B	8AH	|XX| | X| | X|
55	10001001B	89H	|XX| | X| |X |
56	10001000B	88H	|XX| | X| |XX|
57	10000111B	87H	|XX| |X | | |
58	10000110B	86H	|XX| |X | | X|
59	10000101B	85H	|XX| |X | |X |
60	10000100B	84H	|XX| |X | |XX|
61	10000011B	83H	|XX| |XX| | |
62	10000010B	82H	|XX| |XX| | X|
63	10000001B	81H	|XX| |XX| |X |
64	10000000B	80H	|XX| |XX| |XX|

WARRANTY

TECHNICAL DESIGN LABS INC., in recognition of its responsibility to provide quality components and adequate instructions for their proper use and assembly, warrants its' products as follows:

All components sold by Technical Design Labs Inc., (hereinafter referred to as TDL) are first quality prime and are procured from reputable distributors and/or factories and their representatives, and any part which fails because of defects in manufacture or material will be replaced at no charge for a period of 3 months for assembled products following the date of purchase measured from the date of receipt. For replacement, the defective part must be returned to TDL postpaid within the warranty period.

Any malfunctioning unit or subunit returned to TDL within this 3 month warranty period, which in the judgement of TDL has not been subject to electrical or mechanical abuse, will be restored to proper operating condition or replaced at TDL's discretion and returned, with a minimal charge to cover postage.

Any units or subunits purchased and returned to TDL within the 3 month warranty period, which in the opinion of TDL is not covered by the above conditions will be repaired and returned at a cost commensurate with the work required. In no case will this charge exceed $30.00 without prior notification and approval of the owner.

Any unit or subunit, purchased as assembled units are guaranteed to meet the specifications in effect at the time of manufacture for a period of 90 days following purchase.

This warranty is made in lieu of all other warrantie expressed or implied and is limited in any case to the repair or replacement of the unit or subunit involved.

READER'S COMMENTS

TDL
VDB PRELIMINARY MANUAL

In a constant effort to improve the quality and usefulness of its publications, Technical Design Labs, Inc. provides this page for user feedback. Your critical evaluations of this document is our only effective means of determining its serviceability. Please give specific page and line references where applicable.

ERRORS NOTED IN THIS PUBLICATION:

SUGGESTIONS FOR IMPROVING THIS PUBLICATION: (i.e. clarity, organization,convenience,accuracy,legibility.)

MISSING DOCUMENTATION: (i.e. completeness.)

Name----------Date----------

Street----------

City----------State----------Zip Code----------

All comments and suggestions become the property of TDL. Send to Technical Design Labs, Inc.
Dept. of Product Improvements
1101 State Road
Princeton, N. J. 08540

Please indicate in the space below if you wish a reply.
