

---

# Technical Manual

## Hardware Z-100 Series Computers

593-0038-04  
CONSISTS OF

MANUAL  
595-2918-04

FLYSHEET  
597-2792-04

TAB SET (VOL. I)  
597-3437

TAB SET (VOL. II)  
597-3438

SCHEMATIC ENVELOPES  
597-2918-02

MAIN BOARD SCHEMATIC  
585-0018-02

VIDEO LOGIC SCHEMATIC  
585-0019-01

VIDEO DEFLECTION SCHEMATIC  
585-0020-01

FLOPPY CONTROLLER SCHEMATIC  
585-0021-02

TM-100



**This Document was scanned and  
contributed by:**

**Barry A. Watzman**

---

# Video Logic Board

Description .....	4.2
User Options and Jumpers .....	4.3
Theory of Operation .....	4.5
Programming Data .....	4.29
Circuit Description .....	4.48
Troubleshooting .....	4.69
Replacement Parts List .....	4.70
Semiconductor Identification .....	4.72
Circuit Board X-Ray View .....	4.106
Interconnect Pin Definitions .....	4.107
Schematic .....	(Inside Envelope at rear of manual)

## DESCRIPTION

---

The video logic board produces video signals for an internal (in the All-In-One model) or external (Low-Profile model) video monitor. (An external monitor is also used with the All-In-One model for color displays.)

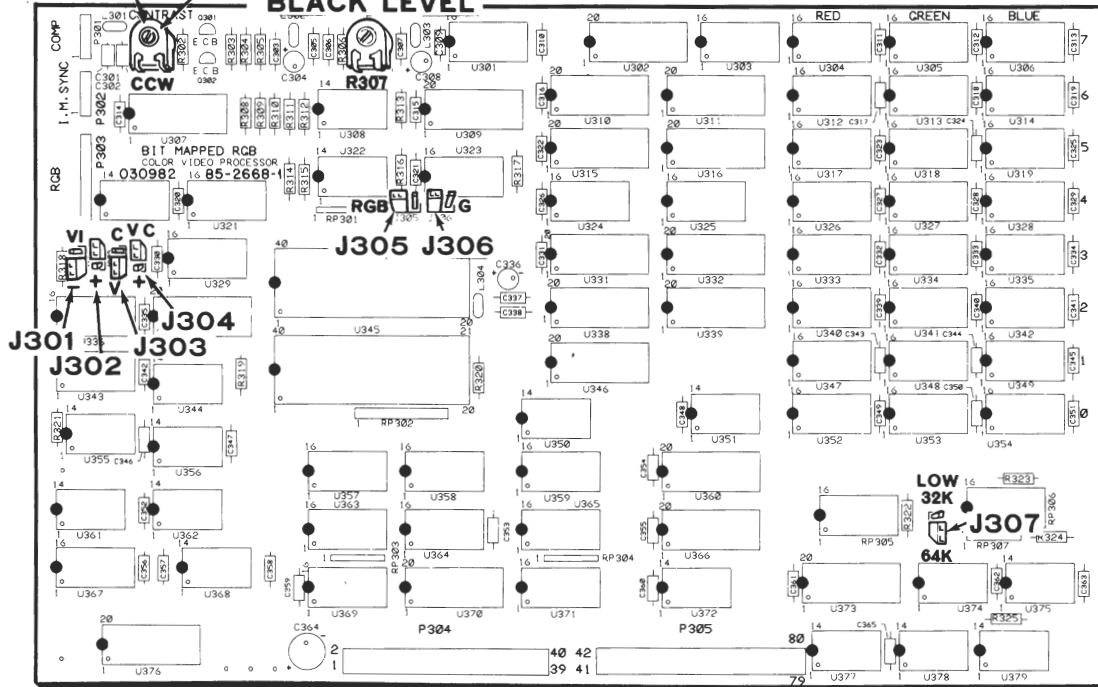
Signals available include composite monochrome video, composite sync, horizontal sync, vertical sync, and three planes of noncomposite video for use with RGB type color monitors.

The normal display format is 25 rows of 80 characters, with each character consisting of an 8 wide by 9 high character cell. However, as the video board uses bit-mapped pixel display technology, character cell boundaries are arbitrary and the display may be more generally thought of as a 640 × 225 pixel graphics display. Also, the video board may be programmed for nonstandard alternate formats, including interlaced displays of up to 640 × 525 pixels or multipage displays. Some nonstandard formats will require 64K video memory chips.

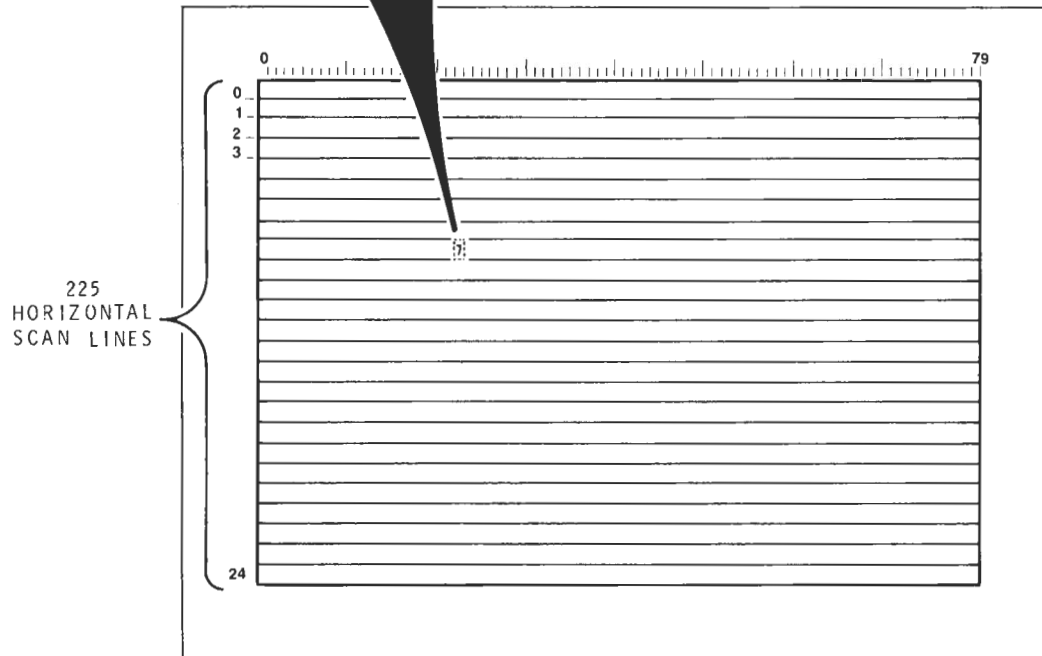
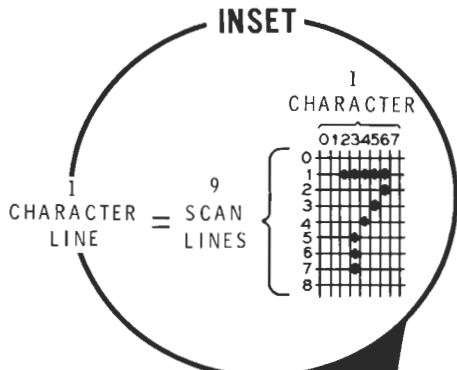
THIS PART MAY NOT  
BE IN ALL UNITS.

CONTRAST

BLACK LEVEL



**PICTORIAL 4-1**  
**Video Logic Circuit Board**



**PICTORIAL 4-2**  
Video Display

## USER OPTIONS AND JUMPERS

---

Refer to Pictorial 4-1 as you read the following information.

### Circuit Board Jumpers

The video logic circuit board jumpers perform the following functions:

- J301 — Selects the polarity of the vertical sync signal for the internal monitor. Putting the jumper on the “-” marked side selects negative polarity. This is its normal position.
- J302 — Selects the polarity of the horizontal sync signal for an external RGB monitor. Placing the jumper on the “+” marked side selects positive polarity. H is the normal position.
- J303 — Selects either composite sync or vertical sync for the external RGB monitor. Placing the jumper on the “V” marked side selects vertical sync. This is the normal position.
- J304 — Selects the polarity of the synchronization signal selected by J303. V/C is the normal position.
- J305 & J306 — These jumpers select color or black and white video. For color, both jumpers must be on the side marked “RGB.” For monochrome, both jumpers must be on the side marked “G.” When you are using color, all three RAM banks are enabled and must have RAMs installed. For monochrome, only the green bank is used.

## USER OPTIONS AND JUMPERS

---

J307 — This jumper allows for different types of RAM to be used.

1. If the jumper is placed on the side marked "LOW 32K", lower 32K type RAM chips are selected.
2. If the jumper is placed on the side marked "64K", 64K type RAM chips are selected.
3. If no jumper is installed, upper-type 32K RAM chips are selected.

### **Black Level Control**

This control (R307) should be set initially at the 1 o'clock position, as shown, and then adjust (if necessary) for a desired display. You do not need to readjust this control if you are using a monitor that has its own black level control.

### **Contrast Control**

Set this control (R301, not installed on all units) fully counterclockwise.



# THEORY OF OPERATION

---

## General Theory

The video logic board signals produce 25 lines of characters on the display screen with 80 characters per line. The board also controls the display colors or gray scales, depending on whether a color or monochrome display is used, and it contains the light pen circuitry.

NOTE: In the following description, the 25 character lines are numbered 0 through 24 and the 80 characters per line are numbered 0 through 79.

### Matrix Scheme

Pictorial 4-2 (Fold-out from Page 4.2) shows the 225 horizontal **scan** lines, produced by the video deflection circuits, that make up the video display on the screen. These 225 scan lines are logically grouped so that every nine scan lines function together to produce one **character** line. (See the inset drawing.) The result of this grouping is 25 character lines on the screen ( $225/9 = 25$ ).

Each of the 25 character lines can display 80 characters. As shown in the inset drawing, each character is made up of 72 dots (called pixels) from an  $8 \times 9$  pixel matrix ( $8 \times 9 = 72$ ). The character that is displayed depends on which pixels are turned on. In the inset drawing, the proper pixels are turned on to display the number 7.

Each pixel has an address in memory and can be turned on individually. Font tables, which define the shape of each character, are contained in the ROM and are down-loaded into system RAM during the boot sequence. Each character in the font consists of nine 8-bit bytes of data ( $8 \times 9 = 72$  bits). Therefore, by changing a character's font data bytes, a character can be redefined to any one of 2 to the 72nd power character shapes.

The present font is arranged as shown in the ASCII chart in the "Programming Data" section of this Manual (Page 10.31).

# THEORY OF OPERATION

---

## General Theory

After a keyboard key is pressed ("7", for example) and software determines that it is time to display the character, the main microprocessor obtains the nine bytes of data that define the character's shape from the "7" entry in the font table and places these nine data bytes in proper locations in video memory. [The memory locations to modify are a function of which character row (0-24) and column (0-79) the "7" is to appear at.] Then, when the display scan lines are refreshed by reading the contents of video memory, the character will be properly displayed on the screen along with any other characters that have been entered.

## Color Display

To produce color, a separate memory plane (array) of video RAM is used for each of the three main colors: red, green, and blue. All the bytes of video RAM that describe a particular color are organized sequentially in 64K (or possibly 32K) byte pages of RAM. The pixel seen on the screen is essentially composed of three superimposed pixels, one in each color plane. Since each of the three color pixels may be on or off, eight different colors are possible. The colors and how they are generated is as follows:

- 0 — That color pixel is off
- 1 — That color pixel is on

<u>Green</u>	<u>Red</u>	<u>Blue</u>	
0	0	0	— Black, no pixels on
0	0	1	— Blue
0	1	0	— Red
0	1	1	— Magenta
1	0	0	— Green
1	0	1	— Cyan
1	1	0	— Yellow
1	1	1	— White

# THEORY OF OPERATION

---

## General Theory

If you only want monochrome, you need only one of the three memory planes. Green is used because the green gun is set for greatest intensity for proper color displays.

If a monochrome display is used with the three memory planes, eight levels of intensity (brightness) can be produced, which corresponds to the above colors. White is the most intense, and black is the least intense.

### **Light Pen**

The light pen is a light detector rather than a light generator. When the pen is turned on and held against the screen, it produces a pulse when the first pixel within its scope is turned on. When this pulse is generated, the present byte address is saved and decoded, and the precise pixel location is remembered.

## THEORY OF OPERATION

---

### Detailed Theory

When software determines that it is time to display a character, the main processor (8088) obtains nine bytes of data that define the character's shape from the font table and places the bytes in proper locations in video memory (VRAM). Then, when the display scan lines are refreshed by reading out of video memory, the character will be properly displayed on the screen.

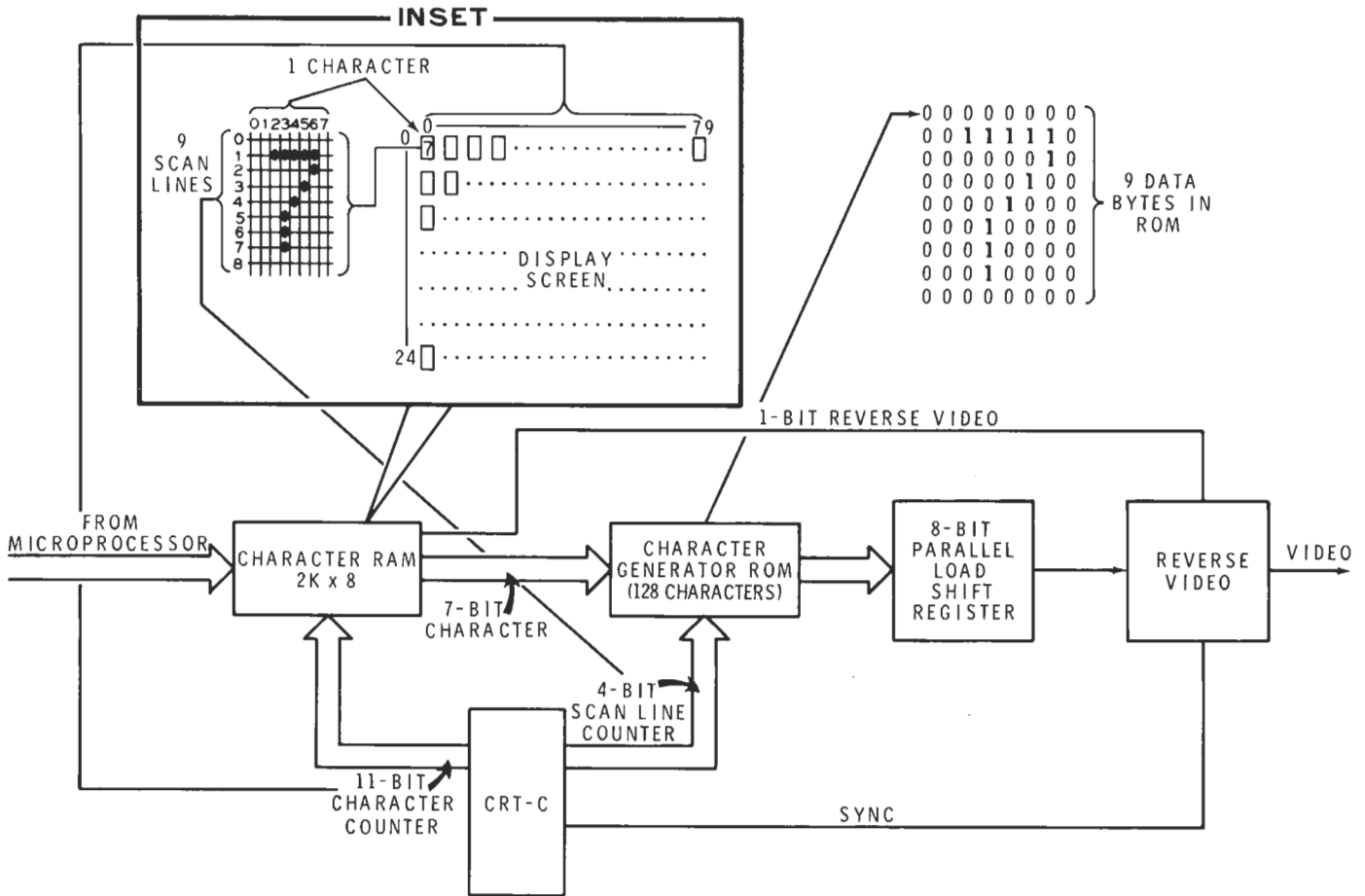
The following pages describe, in detail, how the above video logic functions are performed. The description will first discuss a simple character - generator based video system and then build into the more complex bit-mapped system used in the Z-100 family of Computers.

#### Basic Video System

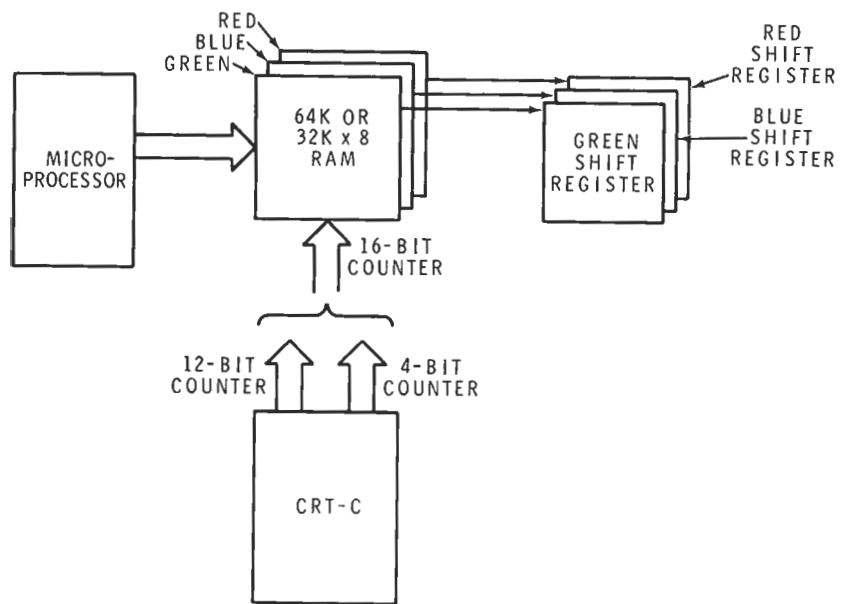
Refer to Pictorial 4-3. The Pictorial shows a basic video system for a 25-character line display that has 9 scan lines per character line and 80 characters per character line.

The microprocessor places a byte of data in the  $2K \times 8$  character RAM for each character on the screen ( $80 \times 25 = 2000$ ). Each data byte represents one of the 128 characters in the character generator ROM, where each of the 128 characters is itself defined by nine data bytes.

The CRT controller (CRT-C) systematically interrogates the character RAM with its 11-bit character counter. It selects character after character until all 80 characters in the row have been selected. During this time, the CRT-C 4-bit scan line counter has selected the first scan line. As each of the 80 characters in RAM is selected, its 7-bit code selects the proper character in the character generator ROM. That character's top byte (part of the first scan line) is shifted out to the video deflection circuits and displayed on the screen. The CRT-C selects these 80 characters and the first scan line is displayed.



**PICTORIAL 4-3**  
Simple System Block Diagram



**PICTORIAL 4-4**  
**Add Memory And Counters**

# THEORY OF OPERATION

---

## Detailed Theory

Then the CRT-C selects these same 80 characters again and again, as the 4-bit scan line counter selects one scan line after another until all nine scan lines of the first character row are displayed. Then the 4-bit scan line counter starts over and the 11-bit character counter selects the next character row in memory. This continues until all the character rows have been selected and the screen is fully displayed.

If reverse video has been selected, then the complement of each bit is sent to the video generation circuits.

## THEORY OF OPERATION

---

### Detailed Theory

#### **Conversion From Character-Based To Pixel-Based Display**

Refer to Pictorial 4-4 (Fold-out from Page 4.8). If we combine the 4-bit scan line address with the 12-bit video refresh address, then a 16-bit address is produced as shown. Also, if we increase the RAM size to 32 kilobytes, we have enough memory to store not only every character ( $80 \times 25 = 2000$  bytes), but all nine bytes of every character ( $80 \times 25 \times 9 = 18000$  bytes).

When a program prints a character, all nine bytes of the character's font pattern are looked up in memory and stored in the  $32K \times 8$  video RAM. Color is achieved by superimposing two other  $32K \times 8$  RAM "pages." These three pages of RAM are used for the red, green, and blue colors. If 64K RAMs are used, the contents of two screens can be placed in memory. As shown, each section of RAM has its own shift register.

Notice that the microprocessor is now connected directly to video RAM. This is so it can write data into the RAM, read data out of the RAM, and select various RAM options that are discussed later.



# THEORY OF OPERATION

---

## Actual Theory

The following paragraphs describe the actual video operation of the Z-100 family of Computers. As shown in Pictorial 4-5, the video logic board consists of:

- A CRT controller.
- A video RAM mapping module.
- Three video RAM planes (arrays).
- Light pen circuitry.

The video RAM mapping module receives addresses from the 8088 microprocessor and changes these addresses into actual video RAM addresses. The address change is done to simplify software.

As shown in Pictorial 4-6, one of the sections of CPU address memory that is not used is "Area B." The RAM mapping module changes the CPU addresses into a more compact sequence such that only the "Displayed Area" data of Pictorial 4-6 is placed in video RAM and the nondisplayed area, "Area B," is ignored.

In Pictorial 4-5, video information is shifted out of video RAM and sent to the video deflection circuits or monitor while the CRT-C sends the sync and timing signals. These RAM signals consist of the data to be displayed on the screen and the sync and timing signals that are necessary to start new scan lines.

The video RAM planes consists of 32 or 64 kilobytes of RAM. The RAM holds one or two screens (with 64K parts) of data that is shifted out and displayed on the screen.

# THEORY OF OPERATION

---

## Actual Theory

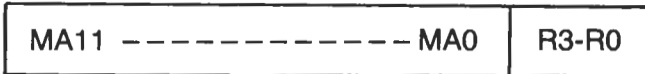
### CRT-C (CRT Controller)

The screen is updated 60 times per second (when set for 60 Hz), with data (characters) from video RAM. During a sweep of the display beam, the CRT-C generates video RAM address VRAMA2 (see Pictorial 4-5) and reads a byte representing eight pixels. Once these pixels are displayed, the CRT-C automatically advances to the next byte that describes the next group of eight pixels. This process continues until the scan line is completed.

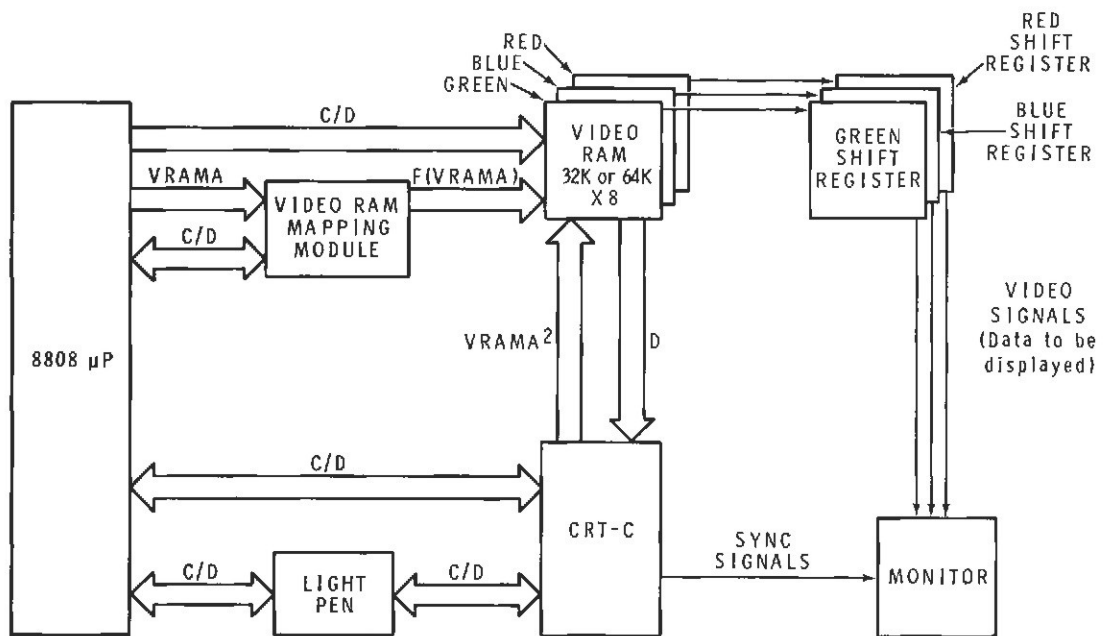
Each byte of video RAM represents eight pixels on the display. Three superimposed bytes are required to fully specify a color pixel. However, for now, consider each pixel to be simply monochrome.

The video RAM address (VRAMA2) is a 16-bit address. Bits 0 through 3, called R3-R0, make up the scan line counter and bits 4 through 15, the memory refresh address MA11-MA0, select the bytes that make up the scan line when the screen is refreshed.

The organization of these scan lines and memory refresh lines, as seen by the CRT-C controller, is crucial to understanding the memory organization. They are organized as follows:

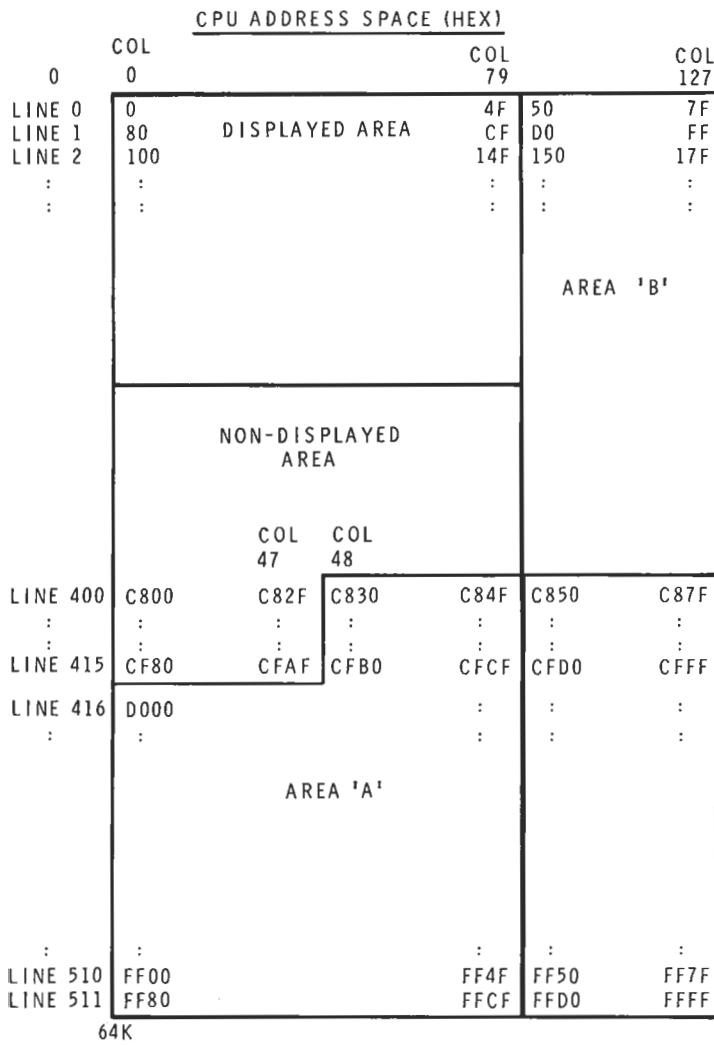


This 16-bit address is presented directly to the video RAM. The CRT-C increments the memory refresh address (MA11-MA0) from the first character address of the line to the last character address of the line for the first scan line. The scan line address, R3-R0, is then incremented. The memory refresh address is then incremented once again from that same first character address to the final character address for the second scan line. This is repeated until all nine scan lines for the character line have been displayed.



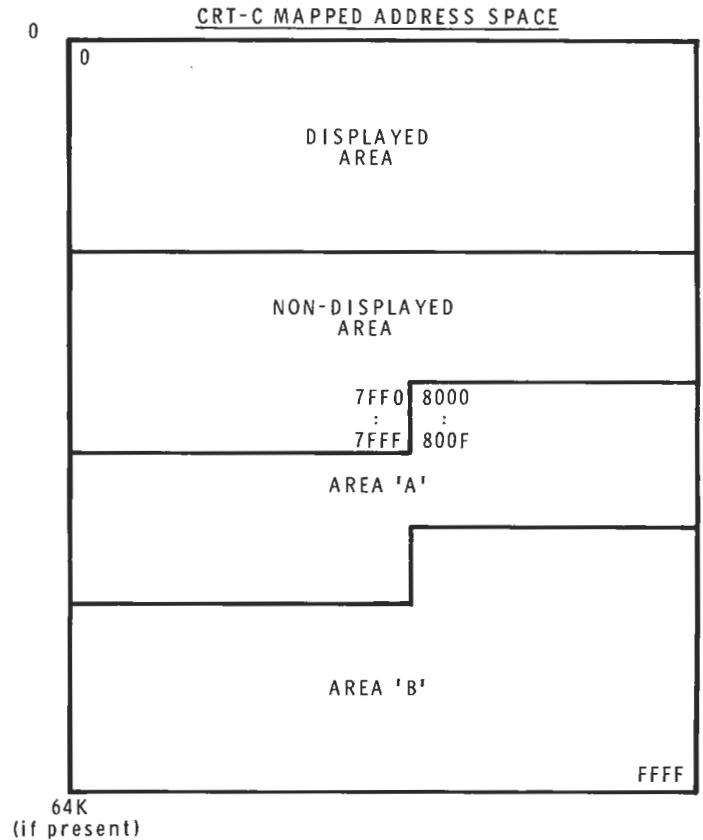
C/D = CONTROL/DATA LINES  
 D = DATA LINES  
 f(x) = FUNCTION OF (X)  
 VRAMA = VIDEO RAM ADDRESS

**PICTORIAL 4-5**  
**Video Block Diagram**



**NOTES:**

1. The boundary between the displayed and the non-displayed areas varies in address depending on the screen mode (9 lines per character, 16 line graphics, etc.).
2. In a system with 32K RAM's installed, areas 'A' and 'B' wrap back into the displayed 32K space.
3. In a system with 64K RAM's installed, areas 'A' and 'B' are addressed by setting the address latch to 80H (assuming the start address is 00).



**PICTORIAL 4-6**  
Video Memory Layout

# THEORY OF OPERATION

## Actual Theory

Once all the scan lines for the character line have been sequenced through, the base address for the memory refresh address (MA11-MA0) is advanced to the first character of the next line. This process is repeated until all lines have been displayed on the screen, at which time vertical retrace takes place. After retrace, the memory refresh address is reinitialized to its start address, and the process repeats.

The CRT-C is programmed for nine scan lines per character, 80 characters per line, and 25 character lines per screen. The addresses shown in Pictorial 4-7 are generated by the CRT-C for each given group of eight pixels (byte of video RAM).

	1st Char Column	2nd Char Column	3rd Char Column	80th Char Column
1st Char, 1st Pixel Row	0	16	32.....	1264
1st Char, 2nd Pixel Row	1	17	33.....	1265
1st Char, 3rd Pixel Row	2	18	34.....	1266
1st Char, 4th Pixel Row	3	19	35.....	1267
1st Char, 5th Pixel Row	4	20	36.....	1268
1st Char, 6th Pixel Row	5	21	37.....	1269
1st Char, 7th Pixel Row	6	22	38.....	1270
1st Char, 8th Pixel Row	7	23	39.....	1271
1st Char, 9th Pixel Row	8	24	40.....	1272
2nd Char, 1st Pixel Row	1280	1296	.....	2544
	1281	1297	.....	2545
	1282	1298	.....	2546
	1288	1304	.....	2552
3rd Char, 1st Pixel Row	2560	.....	.....	3824
25th Char, 1st Pixel Row	30720	30736	.....	31984
25th Char, 9th Pixel Row	30728	30744	.....	31992
-----				
1st character	0*16	1*16	2*16 .....	79*16
	0*16+1	1*16+1	2*16+1.....	79*16+1
	0*16+2	1*16+2	2*16+2.....	79*16+2
	0*16+3	1*16+3	2*16+3.....	79*16+3
	0*16+4	1*16+4	2*16+4.....	79*16+4
	0*16+5	1*16+5	2*16+5.....	79*16+5
	0*16+6	1*16+6	2*16+6.....	79*16+6
	0*16+7	1*16+7	2*16+7.....	79*16+7
	0*16+8	1*16+8	2*16+8.....	79*16+8

**PICTORIAL 4-7**  
CRT-C Memory Addressing

## THEORY OF OPERATION

---

### Actual Theory

In general (assuming the start address is 0), the address of byte “c”, scan line “s”, and row “r” would be:

$$r \times 80 + c \times 16 + s$$

row r,  $0 \leq r \leq 24$

scanline s,  $0 \leq s \leq 8$

character c,  $0 \leq c \leq 79$

One way to scroll the text on this bit-mapped video system would be to move the bytes from one location to another. By moving each byte to the address 128 bytes lower than itself, the entire screen would be effectively scrolled one scan line. (The last scan line should be zeroed to avoid displaying incorrectly initialized memory.) However, this method is not used because of insufficient microprocessor speed and screen ripple.

Scrolling is achieved by adding 1280 bytes ( $80 \times 16$ ) to the start address. The CRT-C begins refreshing the screen from what would normally be the second character line, but displays those characters on the first character line. If the CRT-C parameters have not been changed, an additional line will be displayed at the bottom of the screen as scrolling occurs. Normally, the bottom line is zeroed by the microprocessor during vertical retrace before the start address is advanced. This keeps uninitialized data from being displayed.

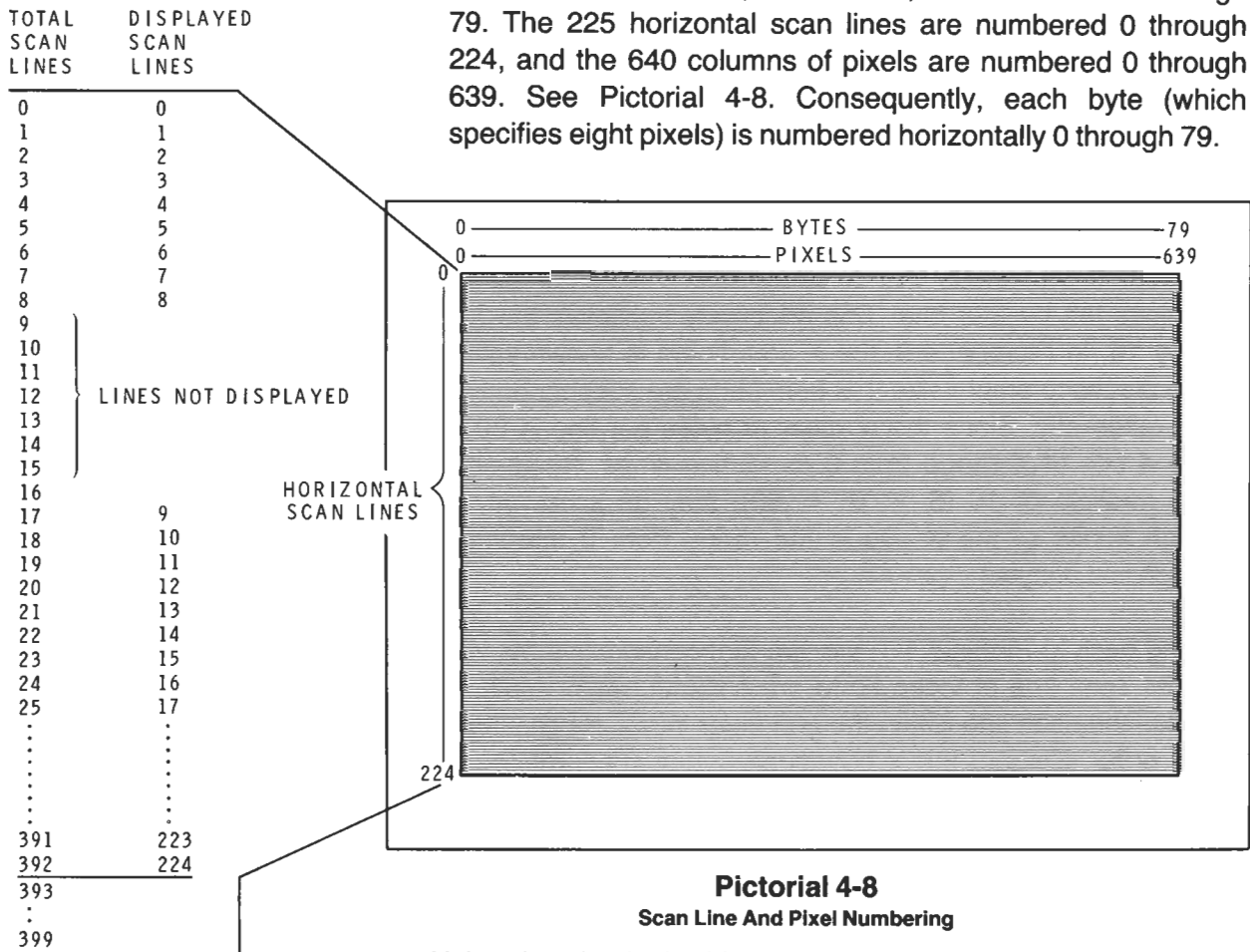
Because the start register is modified during vertical retrace, no characters are displayed at this time, which avoids a momentarily jumbled screen generated from a partially updated start address. To provide vertical synchronization, a video board interrupt is generated.

# THEORY OF OPERATION

## Actual Theory

### Video RAM Mapping Module

As shown in Pictorial 4-2, the rows are numbered 0 through 24 and the characters, or columns, are numbered 0 through 79. The 225 horizontal scan lines are numbered 0 through 224, and the 640 columns of pixels are numbered 0 through 639. See Pictorial 4-8. Consequently, each byte (which specifies eight pixels) is numbered horizontally 0 through 79.



Using the physical addresses shown in Pictorials 4-7 and 4-8 would make programming difficult. Therefore, to simplify things, the video RAM mapping module remaps the video RAM as seen by the 8088 so that pixel addresses are constant without regard to scrolling, and appears in the following chart.

	BYTES	
Row 0:	0,1,2 ...	79
Row 1:	128,129,130 ...	79 + 128
Row 2:	0 + (2*128), 1 + (2*128), 2 + (2*128), ...	79 + (2*128)
Row N:	0 + (N*128), 1 + (N*128), 2 + (N*128), ...	79 + (N*128)
Row 391:	0 + (391*128), 1 + (391*128), 2 + (391*128), ...	79 + (391*128)
Row 392:	0 + (392*128), 1 + (391*128), 2 + (392*128), ...	79 + (392*128)

## THEORY OF OPERATION

---

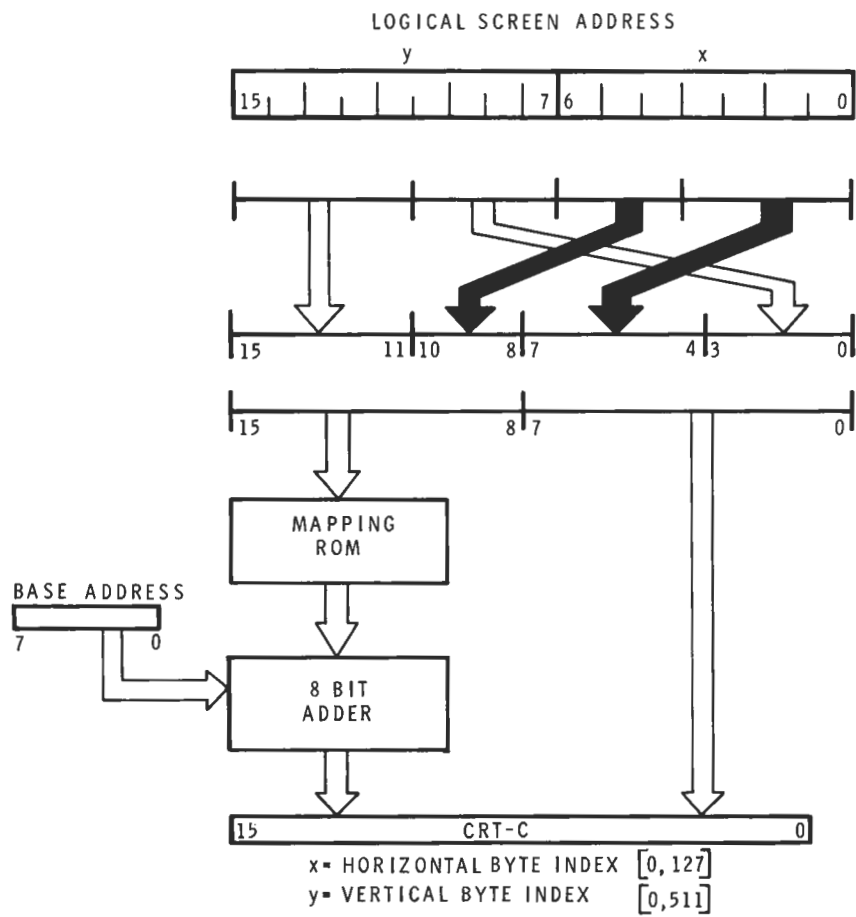
### Actual Theory

Notice that there are “holes” in the addressing map. These holes correspond to the characters 80 - 127 of each row. Since these are illegal character numbers for each row and will not be displayed, you should avoid these addresses. Using them may inadvertently modify pixels of VRAM which you do not intend to modify. Also, notice that whole lines, 9 through 15, 25 through 31, etc., are not displayed. The last displayed line number is 392 (decimal). These nondisplayed line addresses may be used.

The video RAM mapping module reorganizes the video RAM addresses. It first shifts the X coordinate (horizontal byte index) lines left by four bits. This effectively multiplies the X value by 16. In Pictorial 4-9, the X coordinate is shown split into two pieces to emphasize that both parts are subsequently treated differently. Since the number of bytes per line is 80, the low 4 bits of X range from 0 to 15 five times. Because 80 is a multiple of 16 ( $5 \times 16 = 80$ ), they do so evenly. The high 3 bits of the X coordinate range from 0 through 4 for each line of bytes.

Referring back to Pictorial 4-7, we see that consecutive bytes along the scan line of video RAM are consecutive multiples of 16 plus the scan line within the character index. By shifting the low order 4 bits of the Y coordinate (vertical coordinate, that is the scan line) into the low order 4 bits of the output address just vacated by shifting X left, we effectively add in the “scan line counter” component of the video RAM address as generated by the CRT-C.





**PICTORIAL 4-9**  
**Video RAM Mapping Module**

## THEORY OF OPERATION

## Actual Theory

Remembering that X is less than 80, you see that the high address byte does not “sequence” nicely. As sequential horizontal byte addresses are generated for each scan line, the values being generated for the high byte of the address (before they enter the mapping ROM) are:

X'	Y'	A	ROM(A)
0 0 0 0 0:0 0 0	0,0	00H	00H 0,0
0 0 0 0 0:0 0 1	0,1	01H	01H 0,1
0 0 0 0 0:0 1 0	0,2	02H	02H 0,2
0 0 0 0 0:0 1 1	0,3	03H	03H 0,3
0 0 0 0 0:1 0 0	0,4	04H	04H 0,4
0 0 0 0 1:0 0 0	1,0	08H	05H 0,5
0 0 0 0 1:0 0 1	1,1	09H	06H 0,6
0 0 0 0 1:0 1 0	1,2	0AH	07H 0,7
0 0 0 0 1:0 1 1	1,3	0BH	08H 1,0
0 0 0 0 1:1 0 0	1,4	0CH	09H 1,1
0 0 0 1 0:0 0 0	2,0	10H	0AH 1,2
0 0 0 1 0:0 0 1	2,1	11H	0BH 1,3
0 0 0 1 0:0 1 0	2,2	12H	0CH 1,4
0 0 0 1 0:0 1 1	2,3	13H	0DH 1,5
0 0 0 1 0:1 0 0	2,4	14H	0EH 1,6
0 0 0 1 1:0 0 0	3,0	18H	0FH 1,7

The mapping ROM converts one steadily increasing sequence of addresses into a more compact sequence of similar increasing addresses. The mapping ROM takes the data value presented at its input address and outputs the 8-bit value found that corresponds to this internal address. In this way, the “holes” in the logical address space are removed.

# THEORY OF OPERATION

---

## Actual Theory

Several mapping samples are shown below.

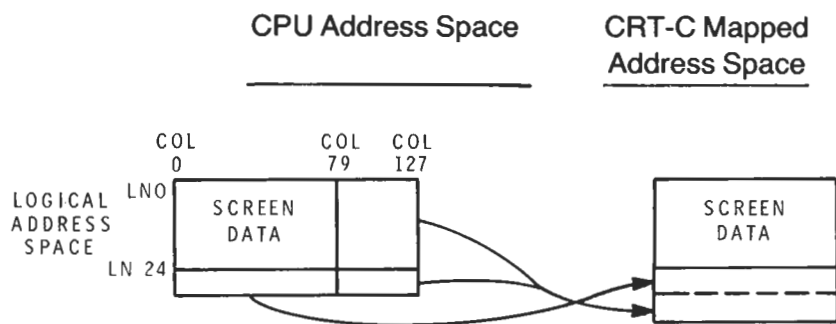
Y (row index)	X (column index)	VRAMA1 address	F(VRAMA1)
0	0	0000H	0000H
0	1	0001H	0010H
0	2	0002H	0020H
0	79	004FH	04F0H
1	0	0080H	0001H
1	1	0081H	0011H
1	2	0082H	0021H
4	0	0200H	0004H
15	0	0780H	000FH
15	79	07CFH	04FFH
16	0	0800H	0500H

The above table shows the address **output** based on the value input. It also shows that once all the legal input addresses have been assigned to their corresponding sequential output addresses, the remainder of the physical, or CRT-C RAM, addresses are assigned sequentially to the logical holes. Since 50H is the first illegal value for the byte of the logical address, it is assigned the address of the first illegal CRT-C address (based on the assumption of 512 lines of 80 bytes). Note that these addresses are assigned sequentially in columns 5-7 and D-F much as the logicals are assigned in columns 0-4 and 8-C.

# THEORY OF OPERATION

## Actual Theory

The full address generated out of the mapping ROM reorganizes the conventional notation into the desired CRT-C address. The 8-bit adder is used in scrolling the screen by advancing the start address. Once the start address is advanced, the data representing the line on the screen is in a different physical address. It is in the place where the old representation for character line 2 was (since it, in fact, is the old representation for character line 2). It is important to understand that data itself does not move. For this reason, the last line on the screen may be in what now is "scrambled" memory. Refer to Pictorials 4-6 and 4-9 and consider the following:



Once the start address is advanced, the new boundaries are as follows:



The data that represents the last line is in a scrambled data area. For this reason, the adder was added to translate the physical addresses.

## THEORY OF OPERATION

---

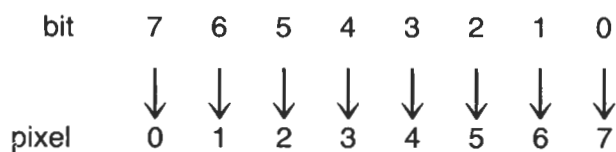
### Actual Theory

By translating the physical address, it is possible to move the mapping function along so that it operates on consecutive "lines" of RAM. The use of this adder is analagous to that of a magnifying glass. The magnifying glass makes a small portion of text easier to read by enlarging the print. The video RAM mapping module has, up to this point, made video RAM easier to address. To reference data outside the area of the magnifying glass, it is simply moved to the new data area. Once the screen image has been "effectively" moved, as is essentially done when the start address of the CRT-C is advanced, the mapping function must be moved.

When the adder is correctly initialized and maintained, the bytes representing line 25 are always in the same **logical** address. This frees the software routines from maintaining a pointer and indexing into the screen data based on the start address. All references to a particular line of data are fixed with respect to the 8088. Furthermore, since the video RAM mapping module is between the 8088 and video RAM, its operation/maintenance do not affect screen refresh in any way. It must only be maintained for easy references to the CRT-C RAM from the CPU.

Pictorial 4-9 shows an 8-bit adder. This adder has been discussed as though it were a full 16-bit adder. In concept, it is. An 8-bit adder is sufficient, however, because the value to be added to this address will always be a multiple of  $80 \times 16 = 5 \times 16 \times 16 = 5 \times 256$ . In the implementation, adding  $n \times (5 \times 256)$  to the address is equivalent to adding  $n \times 5$  to the high order byte of the address because multiplication by 256 is equivalent to shifting left by eight bits. This does assume that the start address will also be a multiple of  $80 \times 16$ . If the start address is initialized to 0, subsequent scrolling operations can maintain it as a multiple of  $16 \times 80$ .

The bits inside each byte of video RAM are mapped to pixels as follows:



# THEORY OF OPERATION

---

## Actual Theory

This means that the value of the byte to turn on the following "x" marked pixel would be one.

0000000X

Similarly, 85 (55 hex) would turn on the following pixels:

0X0X0X0X

To display the pixel in the upper left-hand corner of the screen, 128 (80 hex) would be stored in address 0 of video RAM. The displayed byte would look like:

X0000000

## THEORY OF OPERATION

---

### Actual Theory

#### Video RAM

Each of the three video planes reside in a distinct 8088 64K byte segment. The green plane is at address E0000H, the red plane is at address D0000H, and the blue plane is at address C0000H. In a monochrome system, the green plane is used because video conventions dictate that this plane be of highest intensity. This provides sufficient intensity levels on the monitor without the sensitivity levels being overadjusted.

The planes are organized in decreasing order of relative intensity, with the highest - green - at the top of the available memory space. (The plane addressed at F0000H is reserved for ROM.)

With a monochrome monitor, eight levels of intensity are available with all the video RAM installed. Each level of intensity corresponds to one of the possible colors previously mentioned; white is brightest and black is darkest, no pixels on. Green is 59% of full luminescence, red is 30% of full luminescence, and blue is 11% of full luminescence. In the generation of intensity levels, the luminescence levels add algebraically so that magenta, being composed of red and blue, is 41% of full luminescence.

Because each intensity level corresponds to a color, intensity levels and colors are identical from a software point of view; color produces intensity levels and intensity levels produces color.

Normally, all three video RAM planes are used to update the screen. However, three bits have been provided to disable the displaying of individual video RAM planes. The displaying of one, two, or all three of the planes can be disabled.

## THEORY OF OPERATION

---

### Actual Theory

Another bit totally disengages **all** planes of video RAM from the CPU. When disengaged, the CPU can neither write to nor read from VRAM. This makes sure that VRAM does not conflict with the boot ROM. Note that though the RAM may be disengaged from the CPU, any enabled planes will be displayed on the monitor. Also, when video RAM is enabled and accessed, the "PHANTOM" line is asserted on the S-100 bus.

The remaining bits associated with the video RAM are all designed to optimize various software functions. The first of these is the "plane write" bits. These allow you to access and modify multiple planes of RAM simultaneously.

There is one write bit for each of the three VRAM planes - red, green, and blue. When all of the bits are set high, writing to any of these planes affects only that plane. However, if the write-green bit is set to zero, any writes to the red or blue planes will be made to the green plane also. But, if only the green plane is written to, only the green plane will be changed. Also, the write bits have no effect on the read operations of any plane.

Write bits can be used to produce colored characters on a black background. Just set the write bits to the appropriate combinations for the desired color and write the character to one of the planes. Note, however, that the pixel patterns of the planes must be identical or this mode may not be used.

Write bits can also clear a screen. If you set all the write bits to zero (active low), all the video planes can be zero'ed at once. Setting the planes to a selected background color requires at most two passes. The first pass might zero all planes, and the second one would write all "ones" to the planes that make up the background color. For example, to set the monitor background color to magenta, set write-green and write-red to zero. Then clear the blue plane by zeroing all the bytes. Next, clear the write-red bit to zero and set write-blue and write-green to one. Writing all "ones" to the blue plane will write these same patterns to the red plane.



## THEORY OF OPERATION

---

### Actual Theory

There are two other bits that control screen functions. These two bits may be used to clear the screen without explicitly modifying each byte of the plane with a memory modification generated by the 8088. These bits do not work in conjunction with the write bits and, in fact, override them. They also override the VRAM enable bit, the FLASH bit, and the three plane enable bits. While the clear screen bit is set, each bit of each byte of video RAM addressed by the CRT controller is set to zero or one, depending on the value of the screen's set polarity bit. All planes are quickly modified by this mode and should be used carefully.

### Light Pen

#### General Theory

The light pen is actually a light detector. It detects light and provides a pulse compatible with the logic used in the Computer. If this pulse is used to latch the position where the light pen hit occurred, the software can modify the corresponding memory location and produce the desired results.

The CRT controller (CRT-C) used in the Z-100 has the provision only to store the character number (relative position of the character where the light pen hit occurred with respect to the top left-hand corner of the screen). External circuitry is provided on the video board to also store the scan line number of the character and the pixel position within that line (byte). Thus, the light pen circuitry is capable of resolving a single pixel within the array of  $640 \times 225$ , or  $640 \times 450$  pixels when in the interlace mode. This assumes, of course, that the light pen used is sensitive enough.

# THEORY OF OPERATION


---

## Actual Theory


### Technical Description

Most light pens, in addition to providing a pulse from a detector, also provide a switch. U134-B (LS14, pins 3 & 4, located on the main circuit board) buffers the switch input and feeds it to U114 pin 8 (PA6 – bit 6 of port A of 68A21). Software can poll this input and detect whether the switch is closed or open, and proceed accordingly.

Light pen connector J4 has three more pins in addition to the switch mentioned above. Two pins are for power; ground and Vcc (+5V). The remaining pin is for the pulse output from the light pen. U134-C (LS14, pins 5 & 6) buffers this signal.

Most pens produce a negative-going pulse. Therefore, inverter U134, pin 6, is used to provide the necessary positive-going pulse . An additional inverter is provided (U134, D) in case the pen generates a positive-going pulse. J103 is a 3-pin jumper which selects the positive- or negative-going pulses.

The positive-going pulse is fed to the clock input of U131 pin 3 (74ALS74). The CLR input of that flip-flop (pin 1) is driven by U114 pin 9 (PA7 – bit 7 of port A). If this bit is cleared to zero, the flip-flop will stay cleared and the Q output will not generate the LTPNSTB (light pen strobe) signal.

Assume that this bit is set to one (1). Since the data input of the flip-flop (pin 2) is tied to +5 volts through a resistor (marked HI1 in the schematic), the flip-flop is ready to be clocked. When a positive-going pulse is fed to the clock input, the Q output will be set (logic high); so the LTPNSTB signal will make a positive transition . Subsequent light pulses will not affect the output until the software clears the flip-flop by toggling its clear input.

## THEORY OF OPERATION

---

### Actual Theory

The LTPNSTB signal is applied to video logic board U362 pin 12, the D input of a 74ALS74 flip-flop. This is synchronized by video clock signals and the final output is taken from U356 pin 9 (the Q output of the flip-flop). The signal is then applied to U330 pin 3 (CRT-C – LPSTB input). When a positive transition occurs on this input, the CRT-C latches the value of MA0-MA13 in the internal registers. (Refer to the Device Data Sheet for HD 68A45 for more details.) This, in essence, is how the CRT-C stores the character position.

The scan line value and a pixel position within a given byte are still needed. U356 pin 5 synchronizes LTPNSTB with video DOT CLK to generate PENSTBD (pen strobe delayed), which is fed to U315 – 74LS374 – an octal flip-flop. Four of its inputs are RA0-RA3. Hence, on the occurrence of PENSTBD, these four bits will be clocked into U315, which can be read out by software. (The information is available on the most significant nibble, D4-D7.) U324 is a counter configured as a “DOWN COUNTER”. Its outputs, D0T0-2, are also fed into octal register U315. Hence, the signal PENSTBD will clock these three bits also. Therefore, U315 will provide the scan line number as well as pixel position within a given byte.

### Software Considerations

For the following discussions, you should first read the description of the mapping of the video memory from the CPU and CRT-C points of view presented earlier in this section.

Once the light pen hit has occurred, the problem is how to find the pixel position on the screen and how to find the corresponding memory location. Refer to the Device Data Sheet for the 68A45 for reading the internal registers. First, read the high and low bytes of the LIGHT PEN ADDRESS REGISTERS (R16 and R17). Then read the START ADDRESS high

# THEORY OF OPERATION

## Actual Theory

& low bytes (R12 & R13), and subtract the latter from the former.

$$\begin{aligned} \text{CHARACTER POSITION} &= \\ &= \text{LIGHT PEN ADDRESS} + \text{START ADDR} \end{aligned}$$

For example, if you get  $01ED_H$  for the LIGHT PEN and  $0140_H$  for the START ADDRESSES, then:

$$\begin{array}{r} \text{Character position} = \quad 01ED_H \\ \quad \quad \quad \quad \quad - 0140_H \\ \hline \quad \quad \quad \quad \quad 0AD_H \end{array}$$

$0AD_H$  (173 decimal) is the character position. A correction factor needs to be applied and we will discuss it later. Since the CRT-C is programmed for 80 characters per row, dividing the character position by 80, the quotient and remainder will tell us the row and the character number within that row where the light pen hit occurred.

Example:  $\frac{0AD_H}{50_H} \quad \frac{173}{80}$

$$\begin{aligned} 02_H &= \text{QUOTIENT} = 2 \\ 0D_H &= \text{REMAINDER} = 13 \end{aligned}$$

	0	1				4F
ROW 0	000	001				04F <sub>H</sub>
ROW 1	050 <sub>H</sub>	051 <sub>H</sub>	0C <sub>H</sub>	0D <sub>H</sub>	0E <sub>H</sub>	09F <sub>H</sub>
ROW 2	0A0 <sub>H</sub>	0A1 <sub>H</sub>	0AD <sub>H</sub>			

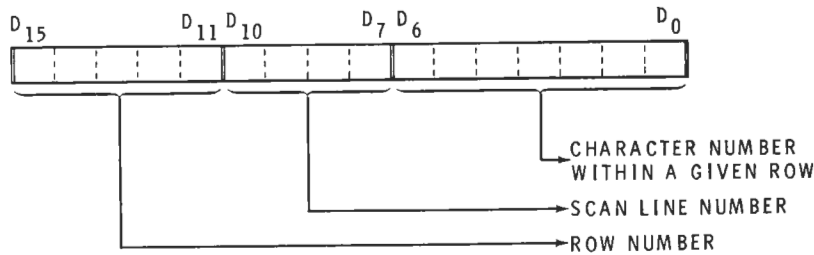
This means the light pen hit occurred on row 2 (3rd row) and character number 13 decimal ( $0D_H$ , 14th character).

Now, by reading the light pen port, one can get the scan line number ( $D_4 - D_7$ ). Let us assume that the value is 6. Now all there is left is to find the corresponding memory location.

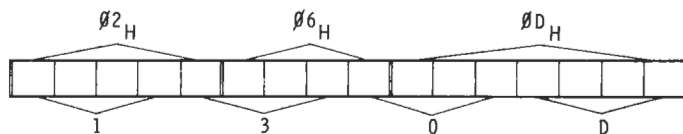
# THEORY OF OPERATION

## Actual Theory

Recall that the 16-bit memory value for VRAM is organized as follows:



In this example, CHAR # = 0D<sub>H</sub>  
 SCAN LINE # = 06<sub>H</sub>  
 ROW # = 02<sub>H</sub>



This translates into memory location  
 130D<sub>H</sub>

Within the byte addressed by 130D<sub>H</sub> the pixel position can be obtained by reading the light pen port bits 0-2.

The above computation assumed that no correction was involved. Recall that we mentioned earlier that a correction is needed to be done on the number we get by subtracting START ADDR from LIGHT PEN ADDR. This is due to the fact that a definite amount of delay is involved in the monitor, light pen, and video circuitry; which is approximately 2 to 5 character times. Therefore, we need to find out for a given system what this correction factor is and then subtract this number from the calculated value. Proceed with the computation of the memory address only after you make this correction. (For example, we might have gotten the character position value as 0B2<sub>H</sub> and apply a correction of 5. Then:

$$0B2_H - 05_H = 0AD_H$$

## PROGRAMMING DATA

---

### Port Addresses

The information in this section concerns the video logic circuit board only and is for the experienced programmer. Programming information for the entire system is contained in "Programming Data" toward the end of this Manual.

The following chart lists the port addresses for devices that are located on the video logic circuit board. A more complete list can be found in the "Programming Data" section in the rear of this Manual.

<u>Device Name</u>	<u>Port Address</u>
6845 CRT-C	DD Register R0–R17
–6845 CRT-C	DC Address register
Video 68A21	DB Control port B
–Video 68A21	DA Address latch
–Video 68A21	D9 Control port A
–Video 68A21	D8 I/O port

## PROGRAMMING DATA

---

### Modifying the Video Control Register

NOTE: It is assumed that the CRT-C (68A45) and the CRT I/O control port (68A21) have been correctly initialized.

The I/O port address for this control port is D8 hex.

The upper four bits (D7 – D4) control the CPU access of video memory (VRAM). The lower four bits (D3 – D0) have nothing to do with VRAM access, but instead control what is displayed on the screen. That is, they control the data coming out of the VRAM that is applied to the pixel control logic. It should be emphasized that the most significant nibble and least significant nibble of the control port D8 (hex) have **no** mutual interaction; control of CPU access of VRAM is independent of control of video display.

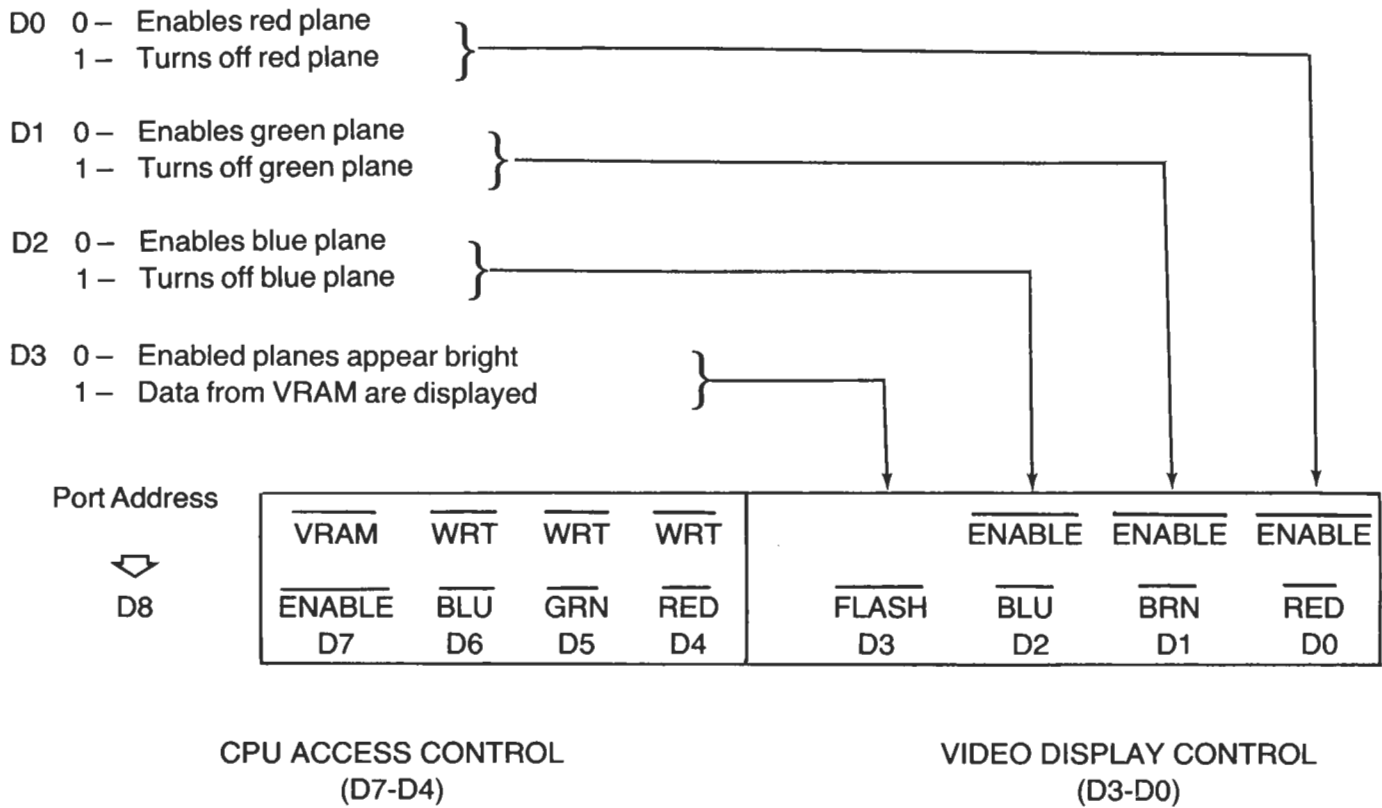
All bits are active low; zero “0” is TRUE and one “1” is FALSE.

# PROGRAMMING DATA

## Modifying the Video Control Register

The memory map of the three planes is as follows. Addresses are in hex.

Green: 0E0000 -- OEFFFF = 64K  
 Red: 0D0000 -- ODFFFF = 64K  
 Blue: 0C0000 -- OCFFFF = 64K



- D7 0 - VRAM is ENABLED. This is the normal operating mode.
- 1 - VRAM is TURNED OFF. The VRAM cannot be accessed at all when turned off.



## PROGRAMMING DATA

---

### Modifying the Video Control Register

The next three bits control “simultaneous write” capability.

D6 0 – When data is written into any color (R, G, or B VRAM), the same data is also written into blue VRAM.

1 – Data can be written into blue VRAM only if the blue plane is accessed.

D5 & D4 – Similarly controls the green and red VRAM.

#### Port D8 Video Display Control Bits (D3-D0)

The following chart shows how the video display control bits (D3-D0) of port D8 can control the screen display.

D3	D2	D1	D0	
FLASH	$\overline{\text{EN}}_{\text{BLU}}$	$\overline{\text{EN}}_{\text{GRN}}$	$\overline{\text{EN}}_{\text{RED}}$	
0	0	0	0	— The screen appears white no matter what VRAM contains.
0	0	0	1	— The screen appears cyan no matter what VRAM contains.
0	0	1	0	— The screen appears magenta no matter what VRAM contains.
0	0	1	1	— The screen appears blue no matter what VRAM contains.
0	1	0	0	— The screen appears yellow no matter what VRAM contains.
0	1	0	1	— The screen appears green no matter what VRAM contains.
0	1	1	0	— The screen appears red no matter what VRAM contains.

## PROGRAMMING DATA

## Modifying the Video Control Register

0	1	1	1	— The screen is blanked (black).
1	0	0	0	— All planes are enabled. VRAM data appears.
1	0	0	1	— Blue and green planes are enabled. VRAM data appears.
1	0	1	0	— Blue and red planes are enabled. VRAM data appears.
1	0	1	1	— Blue plane is enabled. VRAM data appears.
1	1	0	0	— Green and red planes are enabled. VRAM data appears.
1	1	0	1	— Green plane is enabled. VRAM data appears.
1	1	1	0	— Red plane is enabled. VRAM data appears.
1	1	1	1	— The screen is blanked.

## PROGRAMMING DATA

---

### Modifying the Video Control Register

The following are examples of how the screen can be controlled by data bits D3-D0.

#### Example 1:

D3	D2	D1	D0
	$\overline{\text{EN}}$	$\overline{\text{EN}}$	$\overline{\text{EN}}$
$\overline{\text{FLASH}}$	$\overline{\text{BLU}}$	$\overline{\text{GRN}}$	$\overline{\text{RED}}$

0      1      0      1 — D3 is 0, so the flash bit is turned on and VRAM data is masked. Those planes enabled will appear.

D0 & D2 = 1 so the red and blue planes are turned off. D1 = 0 so the green plane is enabled and the screen is green.

#### Example 2:

D3	D2	D1	D0
	$\overline{\text{EN}}$	$\overline{\text{EN}}$	$\overline{\text{EN}}$
$\overline{\text{FLASH}}$	$\overline{\text{BLU}}$	$\overline{\text{GRN}}$	$\overline{\text{RED}}$

1      0      1      0 — D3 is 1, so the flash bit is turned off and the VRAM data will appear on the screen.

D2 & D0 = 0, so the blue and red planes are turned on. D1 = 1, so the green plane is turned off.

## PROGRAMMING DATA

### Modifying the Video Control Register

Actual VRAM data will appear on the screen with the green plane disabled. "Green plane disabled" means that there will be no green pixel turned on. The actual data contained in the green plane's VRAM (0E0000 – 0EFFFF) is unaffected.

The normal operating mode is as follows. All the planes are enabled and the flash bit is turned off.

D3	D2	D1	D0
	$\overline{\text{EN}}$	$\overline{\text{EN}}$	$\overline{\text{EN}}$
$\overline{\text{FLASH}}$	$\overline{\text{BLU}}$	$\overline{\text{GRN}}$	$\overline{\text{RED}}$
1	0	0	0

#### Port D8 CPU Access Control Bits (D7-D4)

Bit D7 is the VRAM ENABLE bit. It is like a master switch. When D7 = 1: VRAM is turned off; D6-D4 have no effect on VRAM access; and the CPU will not be able to read from or write to any plane, red, green, or blue.

When D7 = 0, video RAM is enabled. This is the normal operating mode. Bits D6-D4 control simultaneous write capability. Obviously, the processor can read only one plane at a time, so these bits control only write accesses to VRAM and have no effect on read cycles.

When D6 = 0, it enables simultaneous write to blue VRAM when any color VRAM is written into. If the CPU writes to red VRAM or green VRAM (or blue VRAM), blue VRAM is also written into. Note that D6 has no control over other colors. When D6 = 1, this feature is turned off for blue VRAM.

## PROGRAMMING DATA

### Modifying the Video Control Register

In a similar manner, D5 controls green and D4 controls red VRAM.

#### Example 1:

D7	D6	D5	D4	
$\overline{\text{VRAM}}$	$\overline{\text{WR}}$	$\overline{\text{WR}}$	$\overline{\text{WR}}$	
ENABLE	BLU	GRN	RED	
0	1	0	0	—

D7 = 0, so VRAM is enabled.  
D6 = 1, so  $\overline{\text{WRITE BLU}}$  is off.  
D5 & D4 = 0, so  $\overline{\text{WRITE GRN}}$  and  $\overline{\text{WRITE RED}}$  are on.

Suppose that the CPU wants to write 5D (hex) to location 0E68C0 (hex). The CPU, while trying to write to one plane (green), will simultaneously modify two corresponding memory locations in two color planes (red and green).

1. Location 0E68C0 is in the green VRAM. Therefore, no matter what bits D6-D4 are, green VRAM location 0E68C0 will be modified to 5D.
2. D6 = 1. Therefore, the blue VRAM is not affected.
3. D5 = 0. The CPU is writing to the green VRAM and so its  $\overline{\text{WRITE GREEN}}$  bit has no effect.
4. D4 = 0. The  $\overline{\text{WRITE RED}}$  bit has been turned on. Therefore, even though the processor is writing to only the **green** plane, red is also written into. The red plane occupies the address range 0D0000 – 0DFFFF, so data 5D will be written into location 0D68C0 also.

## PROGRAMMING DATA

### Modifying the Video Control Register

---

#### Example 2:

Assume that all three planes of VRAM have been cleared (00). Then suppose we want to write OFF (hex) (a solid line \_\_\_\_\_) on the screen to location 0000 in magenta (red and blue).

To do this:

1. Make sure that D7 of port D8 = 0. This enables VRAM. (This is the default status.)
2. Enable one of the desired plane's write bit, say red. Then D6 = 1, D5 = 1, and D4 = 0.
3. Write to the corresponding location in the other (blue) plane, since blue is in the "C page." Write FF (hex) to location 0C0000 (hex).

The above three steps produce the desired results, but a slightly better scheme avoids the work of keeping track of colors. That is to turn on the bits of the planes we want. So, to do this, perform the second and third steps as follows:

2. Enable the desired plane's write bits. D6 = 0, D5 = 1, D4 = 0.
3. Write FF (hex) to either location 0C0000 (hex) or location 0D0000 (hex).

For alphanumeric applications, like terminal emulation, etc., where the main emphasis would be writing characters in white, the mode would be D7 = 0, D6 = 0, D5 = 0, D4 = 0.

## PROGRAMMING DATA

---

### Modifying the Video Control Register

Typically, all three planes would be displayed. So the I/O port would read:

D8 0 0 0 0 1 0 0 0

port D8 (hex) ==> 08 (hex) ---- alphanumeric mode.

For some graphic applications, where you do not want to write to more than one plane at a time, the value would be

0 1 1 1 1 0 0 0 ==> 78 (hex.)

port D8 (hex) ==> 78 (hex) ---- graphic mode.

These are examples only and are not intended to identify two different modes, graphic and alphanumeric.

## PROGRAMMING DATA

## Modifying the CRT-C Register

The CRT-C (CRT Controller) has an address register AR [port address DC (hex)] and registers R0-R17 [port address DD (hex)].

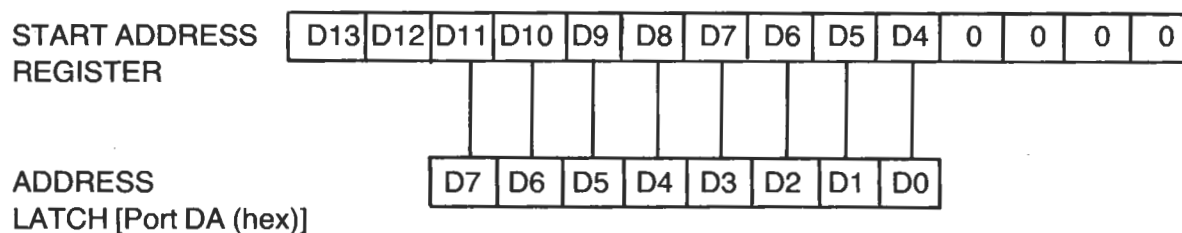
The Address register is a pointer register. It points to one of the 18 registers R0-R17. For example, if you want to access register R12, first write 0C (hex, 12 decimal) into port DC (hex), and then access port DD (hex).

The start address register is 14 bits wide. R12 is the high byte of this register (bits D7 and D6 -- don't care), and R13 is the low byte. R12 and R13 are read/write registers.

Example: Read the low byte of the start address register.

```
MOV AL, 00DH
OUT 0DCH, AL
IN AL, 0DDH
```

In the mapping scheme used in the Z-100 video, an address latch has to be initialized correctly to correspond to the CRT-C's start address register. This address latch is at port DA (hex). In order for the mapping scheme to work, the least significant four bits of the start address register must be zero and bits D4-D11 should match the contents of the address latch.





## PROGRAMMING DATA

---

### Modifying the CRT-C Register

Under normal operating conditions, this will be the case. When in doubt, you should initialize the latch to meet these conditions.

Example:

```

MOV  AL,00CH ; LOAD
OUT  ODCH,AL ; START ADDRESS
IN   AL,ODDH ; HIGH BYTE
MOV  AH,AL   ; IN AH
MOV  AL,00DH ; LOAD
OUT  ODCH,AL ; LOW BYTE
IN   AL,ODDH ; IN AL
MOV  CL,4    ; SHIFT COUNT
SHR  AX,CL   ; AL NOW CONTAINS LATCH VALUE
OUT  ODAH,AL ; INITIALIZE THE LATCH

```

### How to Turn Pixels On and Off

Refer to the "Theory of Operation" on Page 4.5 for a detailed description of how the video section works.

The most significant bit (MSB) of any given byte will be seen on the screen as the left-most pixel and the least significant (LSB) as the right-most pixel. In the following example, 1's indicate turned-on pixels.

```

Example:  1  1  1  0  0  0  0  0 <== SCREEN
          1  1  1  0  0  0  0  0 <== VRAM

```

Since it is straightforward to define the location of a pixel within a given byte, the problem to be solved is to locate the byte in the video memory.

## PROGRAMMING DATA

### Modifying the CRT-C Register

The screen is organized as 640 (decimal) pixels [or 80 (decimal) bytes] horizontally across the screen. To the CPU (or system logical address space), the top left-most byte is always at 0000. (These definitions apply to all planes.) The byte addresses increase from left to right on any given scan line. The line address increases from top to bottom in a given frame. The least significant seven bits define the byte position in a given line, with 00 being the left-most. The most significant nine bits define the line address, with 000 being the top-most line.

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	VRAM ADDR ESS
A6-A0   — Byte address A15-A7  — Line address																

(The following values are in hex.)

	Byte 0	Byte 1	Byte 2	-----	Byte 4F
LN 0	0000	0001	0002	-----	004F
LN 1	0080	0081	0082	-----	00CF
LN 2	0100	0101	0102	-----	014F
LN 3	0180				
LN 4	0200				
LN 5	0280				
LN 6	0300				
LN 7	0380				
LN 8	0400				
-----					
LN 9	0480				
LN A	0500				
LN B	0580				
LN C	0600				
LN D	0680				
LN E	0700				
LN F	0780				
-----					
LN 10	0800	0801	0802	-----	084F
LN 11	0880	0881	0882	-----	08CF

## PROGRAMMING DATA

---

### Modifying the CRT-C Register

At the right-hand side of the screen, past the 80th location, there are “holes” in the logical address space. [See Pictorial 4-2, fold-out from Page 4.2.] You must **not** attempt to use these locations, especially while 32K RAMs are used. For example: never try to write to location 0E0150 (hex) — green plane, line 2, byte 50 (hex).

Depending on whether the CRT-C is programmed for nine scan lines or 16 scan lines per character row, there will or will not be “holes” for entire lines in the CPU’s logical address space. — Holes refer to locations that do not appear. The actual screen will be continuous. For example, when the CRT-C is programmed for 9 scan lines, line 10 (hex) will appear immediately below line 8 (hex).

For terminal emulation applications, where 25 rows of characters need to be displayed, nine scan lines per row are programmed. This is because hardware scrolling is easily done by changing the start address register in the CRT-C, and of course changing the address latch accordingly. On the other hand, where a lot of address computations are involved in some graphics applications, it may be advantageous to disregard the convenience of scrolling and instead choose the continuity in the line numbers (without “holes”).

## PROGRAMMING DATA

### Modifying the CRT-C Register

#### Example 1:

The CRT-C has been programmed for nine scan lines per character row, the CPU writes F0 (hex) to location 0C693F (hex), and 0C (hex) points to the blue plane.

Problem: Figure out the location of 693F (hex) and what is displayed there.

Split 693F (hex) into 9 and 7 bits:

```

693F (hex)
    ==> 0110 1001 0011 1111
    ==> 0 1101 0010 011 1111
    ==> 0D2 (hex)      + 3F (hex)
  
```

3F (hex) denotes the position of the byte across the screen from the left, which is the X coordinate. Therefore, the 693F (hex) location will correspond to the 64th (decimal) byte from the left. In Y coordinate 0D2 (hex), 0D refers to the character row number and 2 refers to the scan line number within the character row. Since the CRT-C has been programmed for nine lines per row of characters, the location will correspond to  $13 \times 9 + 2 = 119$  (decimal).

Data byte F0 (hex) (1 1 1 1 0 0 0 0, turning on the left four pixels) will appear on the 120th line from the top, and 64 bytes from the left, assuming that the proper plane has been enabled.

#### Example 2:

Problem: Access the screen at a location 37 bytes from the left and on the 126th line from the top.

37 bytes from the left translates into byte address 24 (hex). The 126th line means that the intended byte appears on  $(126/9)$  the fourteenth row and the ninth line in that row. So, the address of the line is 0D8 (hex). Remember that the top row is row 0.

## PROGRAMMING DATA

---

### Modifying the CRT-C Register

Therefore, the VRAM location is

$$\begin{array}{rcc} 0D8 \text{ (hex)} + 24 \text{ (hex)} & == > & 6C24 \text{ (hex)} \\ 9 \text{ bits} & & 7 \text{ bits} \end{array}$$

#### Example 3:

Problem: Where on the screen is location 35E3 (hex)?

Divide 35E3 (hex) into its X and Y coordinates.

$$35E3 \text{ (hex)} == > 06B \text{ (hex)} + 63 \text{ (hex)}$$

The X coordinate 63 (hex) falls into the "hole" region of 50 (hex) – 7F (hex). This is prohibited and such an operation should not be attempted.

#### Example 4:

Problem: Where on the screen is location 1727 (hex)?

$$\begin{array}{rcc} 1727 \text{ (hex)} & == > & 02E \text{ (hex)} + 27 \text{ (hex)} \\ & & \text{Y} \quad \quad \text{X} \end{array}$$

The X coordinate is the twenty-eighth byte, and the Y coordinate is the third row and the fifteenth line in that row. Since the CRT-C will display only 9 scan lines, line E (hex), the 15th line will not appear on the screen. If you have a use for these VRAM locations, with scan line addresses 9 (hex) through F (hex), access to these locations is permitted. Remember that only the X coordinate cannot be in the range 50 (hex) through 7F (hex).

When the CRT-C is programmed for 16 scan lines, similar calculations can be made, but now there are **no non-displayable** VRAM locations ("holes" for scan lines 9 through F). Still, the X coordinate restriction **does** apply.

## PROGRAMMING DATA

---

### Clearing the Screen

The “clear screen” feature allows you to initialize all the viewable VRAM locations to either 00 (hex) or FF (hex).

The CRT I/O control port (68A21) has two control ports, A and B. Bit 3 of each port serves a special purpose. Bit 3 of control port A, address D9 (hex), is the  $\overline{\text{CLRSCRN}}$  bit and it is active low. (It is 1 by default.) Bit 3 of control port B, address DB (hex), is the SET bit.

When  $\overline{\text{CLRSCRN}}$  is inactive (1), the SET bit has no effect and normal video operations take place. When  $\overline{\text{CLRSCRN}}$  is programmed to “0”, the SET bit decides whether the VRAM is initialized to 00 (hex) if SET = 0, or to FF (hex) if SET = 1.

Be careful when you use the  $\overline{\text{CLRSCRN}}$  feature. This bit operates independent of all other bits discussed so far. It operates whether or not VRAM is enabled, whether or not multiple write capability is invoked, and whether or not FLASH or any planes are enabled. Activating clear screen can wipe out **all** red, green, and blue VRAM locations.

Ports D9 (hex) and DB (hex), control ports A and B, are readable. Therefore, whenever you want to modify the  $\overline{\text{CLRSCRN}}$  or SET bits, you should first read those ports, modify bit 3, and then write it back.

Notice that even though these bits have complete control over VRAM contents, they still do not affect the video display control bits. For example, assume all three planes were disabled to start with. Now, activating the  $\overline{\text{CLRSCRN}}$  feature with SET = 1 will not make the screen white. The VRAM locations would have been changed to FF (hex).

The  $\overline{\text{CLRSCRN}}$  feature was included because CPU accesses of VRAM are inherently slow (because of arbitration between the CRT-C and the CPU for VRAM access, and the CRT-C has higher priority), and to clear the screen would mean writing to about 20K bytes of memory. The  $\overline{\text{CLRSCRN}}$  feature will clear the screen in 1 frame time, or 16.7 milliseconds for 60 Hz operation.

## PROGRAMMING DATA

---

### Clearing the Screen

#### Example 1:

Clear the screen.

1. Read port D8 (hex) and save the status.
2. Initialize CRT control port D8 (hex) to 0F (hex). This will instantaneously blank the screen, since all three planes are disabled.
3. Write a zero into bit 3 of port DB (hex). This will make SET = 0. (Recall that you have to do the READ, modify, and WRITE sequence.)

```
IN    AL, 0DBH
AND   AL, 0F7H
OUT   0DBH, AL
```

4. Write a zero into bit 3 of port D9 (hex). This will activate CLRSCRN.
5. Wait for 16.7 milliseconds (in the 50 Hz mode, this wait is 20 milliseconds). Do it in either of two ways:
  - A. Use the timer integrated circuit in your system.
  - B. Use the vertical sync pulse. If you have seen two consecutive  $\overline{VSYNC}$  pluses, you know that 16.7 mS have elapsed.
6. Turn off the  $\overline{CLRSCRN}$  bit. (Write a "1" to bit 3, port D9 (hex).)
7. Restore the status to video control port D8 (hex).

## PROGRAMMING DATA

---

### Clearing the Screen

#### Example 2:

Set the screen.

1. Read port D8 (hex) and save the status.
2. Turn on the  $\overline{\text{FLASH}}$  bit and those planes which were originally enabled.

```
IN    AL, 0D8H
AND   AL, 0F7H
OUT   0D8H, AL
```

This will instantaneously set the color of the screen to the planes enabled.

3. Write a "1" into bit 3 of port DB (hex). This will make SET = "1".
4. Write a "0" into bit 3 of port D9 (hex). This will make CLRSCRN = 0.
5. Wait for 16.7 mS (or 20 mS). (See the above example.)
6. Turn off the clear screen bit.
7. Restore the original status to video control port D8 (hex).

Remember: The clear screen feature wipes out all displayable locations, plus some more, on all three VRAM banks.



## CIRCUIT DESCRIPTION

---

### Video Logic Circuit Board

#### Video Processing Circuits

##### Cathode Ray Tube Controller (CRT-C)

The CRT-C, U330, fetches the characters to be displayed and provides horizontal and vertical timing. It also keeps track of the affected character if the light-pen circuits are used.

Briefly, here's what each line does. See the 6845 IC data sheet for more information.

POC Power-on clear, from the S-100 bus. Sets all registers to their initial conditions on power-up or reset.

6845CS Chip-selects the CRT-C for accessing the internal registers.

ECLK Latches the data into or out from the registers on its trailing edge.

BA0 Helps select a specific register inside the CRT-C.

OUT When low, writes data into the selected register. Otherwise, reads data from it.

DIO0-DIO7 Data bus used by the CPU to access the CRT-C registers.

CLK Provides character-clock timing to the CRT-C.

HSYNC Horizontal sync pulse.

VSYNC Vertical sync pulse.

CURSOR Provides an indication where the next character will be printed.

DISEN Disables the display during horizontal and vertical retrace.

# CIRCUIT DESCRIPTION

---

## Video Logic Circuit Board

MA0-MA11 Memory address lines. Point to the current character line, and the character in that line.

RA0-RA3 Row address lines. Points to the current scan line in the current character line.

### Writing to a CRT-C Register

To select a specific CRT-C register (R0-R17), the CPU must first program the address register (AR). For example, to write to R12, the CPU outputs 0CH to port 0DCH. This places the number 12 into AR. The CPU then outputs the data it wants to write to port 0DDH, which is loaded into register 12. Here's how it happens.

The CPU outputs the number 12 (0CH) to port 0DCH. This is coupled through U338 to the data lines of the CRT-C. At this time, 6845CS at VIOSEL (U369) asserts the chip-select line at pin 25 of the CRT-C. Since the port address is 0DCH, line BA0 = 0; thus accessing the AR.

When ECLK goes low, the data (0CH) on the bus lines is loaded into the address register. AR now points to register 12.

The CPU now outputs the byte it wants to write into port 0DDH and line 6845CS is again asserted. Since the port address is 0DDH, line BA0 = 1, telling the CRT-C to route the data to the register pointed to by AR. When ECLK goes low, this data is loaded into register 12.

### Reading Data From The CRT-C

The procedure is the same as writing data, except that U331 is selected instead of U338. This is done by  $\overline{DBIN}$  from the S-100 bus and by  $\overline{4521CS}$  from U366 pin 14.

## CIRCUIT DESCRIPTION

---

### Video Logic Circuit Board

#### How the CRT-C Addresses RAM

As mentioned before, the CRT-C is normally programmed to emulate the H19 video terminal. That is, the display will contain 25 lines, 80 characters per line, and nine scan lines (rows) per character line.

MA0-MA11 points to the character location within a character line, and also points to the current character line. They do this by incrementing the base address by ten after every ten scan lines. RA0-RA3 counts the number of scan lines. After one scan line is complete (MA0-MA11 count to 79), RA0-RA3 reset to 0 and MA0-MA11 reset to their base address. The count begins again. This procedure continues until nine scan lines are processed. RA0-RA3 again returns to zero, but MA0-MA11 increment their base address by ten to point to the next character line.

For each address, a byte is read from video RAM (VRAM) and shifted serially out to the video amplifier with the horizontal and vertical sync pulses. The scan rate is such that each address row appears beneath the previous one so that the serial dots form characters on the screen. Once the last character row is processed, both RA0-RA3 and MA0-MA11 reset to zero, vertical retrace takes place, and the process repeats.

Incidentally, at vertical retrace a sync pulse is sent through U366 (lower left on the schematic) to interrupt the CPU. This permits the CPU to access the CRT-C registers (for example, to scroll the display) without interfering with the display.

The address lines reach memory by passing through a set of multiplexers. RA0-RA3 connects to multiplexer U357 while MA0-MA11 connects to U363, U358, and U359. These ICs allow coupling the CRT-C address lines to VRAM, or the CPU address lines to VRAM.

# CIRCUIT DESCRIPTION

---

## Video Logic Circuit Board

When line VIDRAMSEL is low, the multiplexers pass the CRT-C address bus to the VRAM address bus. RA0-RA11 is the lower 4 bits, DA0-DA3; and MA0-MA11 are bits DA4-DA15. This causes the address line to increment by 16 for every scanned character. See Pictorial 4-10. This shows the on-screen character location and its relative address (decimal) as seen by the CRT-C.

	1st Char <u>Column</u>	2nd Char <u>Column</u>	3rd Char <u>Column</u>	80th Char <u>Column</u>
1st Char, 1st Pixel Row	0	16	32.....	1264
1st Char, 2nd Pixel Row	1	17	33.....	1265
1st Char, 3rd Pixel Row	2	18	34.....	1266
1st Char, 4th Pixel Row	3	19	35.....	1267
1st Char, 5th Pixel Row	4	20	36.....	1268
1st Char, 6th Pixel Row	5	21	37.....	1269
1st Char, 7th Pixel Row	6	22	38.....	1270
1st Char, 8th Pixel Row	7	23	39.....	1271
1st Char, 9th Pixel Row	8	24	40.....	1272
2ndChar, 1st Pixel Row	1280	1296	.....	2544
	1281	1297	.....	2545
	1282	1298	.....	2546
	1288	1304	.....	2552
	2560	.....	.....	3824
25th Char, 1st Pixel Row	30720	30736	.....	31984
25th Char, 9th Pixel Row	30728	30744	.....	31992

**Pictorial 4-10**  
Relative Memory Locations

# CIRCUIT DESCRIPTION

---

## Video Logic Circuit Board

Lines DA0-DA15 connect to address multiplexer U360 and U373. This circuit splits the address for RAS and CAS timing. RAS timing occurs when ADMUX is high, coupling the following lines to the outputs:

```

VA7 VA6 VA5 VA4 VA3 VA2 VA1 VA0
--- --- --- --- --- --- --- ---
DA9 DA8 DA7 DA6 DA5 DA4 DA1 DA2
    
```

CAS timing occurs when ADMUX goes low, causing:

```

VA7 VA6 VA5 VA4 VA3 VA2 VA1 VA0
--- --- --- --- --- --- --- ---
DA15 DA14 DA13 DA12 DA11 DA10 DA0 DA3
    
```

The address lines at VA0 and VA1 are arranged so the RAM ICs can get refreshed during a normal CRT-C scan in both the non-interlace and interlace modes. This results in a reduction of components in the video circuits.

Jumper J307 permits the use of 64K RAMs or 32K RAMs. To use 64K RAMs, connect the jumper from DA15 to U373-11. To use 32K RAMs located in the upper half of the 64K address space, remove the jumper. To use 32K RAMs in the lower half of the 64K address space, connect the jumper from U373-11 to ground. Note that if the computer uses 32K RAMs, they all must be located in either the upper 32K of each 64K bank or all in the lower 32K--they can't be mixed. See the H/Z-100 Memory Map (Pictorial 4-11) located on Page 4.55.

# CIRCUIT DESCRIPTION

## Video Logic Circuit Board

### Converting RAM Data to Video

There are two sets of data lines at the video memory. One is an 8-bit bus, BD0-BD7, used by the CPU to write to RAM; and three 8-bit output buses, one for each color.

The output buses go to the CPU through U339, U310, and U316. Only one of these ICs will be selected to place the data on BDI0-BDI7. This, in turn couples through U223 to the CPU. This will be covered in more detail later.

The three output buses also couple to the video processing circuits through U332, U302, and U311. Here's how the data is processed.

When the CRT-C has control of RAM (which is most of the time, since it has priority), the VRAM is in the read mode. This is due to a logic zero on VIDRAMSEL (U377 pin 4) and CLRSCRN (U366 pin 3). When the addressed data settles, VIDSTRB from U376 pin 17 asserts to latch and RGB data into U332, U302, and U311. (Note: If this is a minimum system - green only - U332 and U311 outputs will remain a steady state.)

Next, the load shift register line from U320 pin 6 goes low to latch the RGB data into the parallel-to-serial converters, U325, U301, and U303. This line pulses at the character clock rate.

The dot clock at pin 6 of these ICs then shifts the data out through pin 13. This takes place at eight times the character clock rate, or 14.112 MHz, which is the rate of the dot clock. While the video information is being shifted out, VIDSTRB is loading the next byte into D latches. When the last dot is shifted out of the parallel-in/serial-out converter, the bytes in the D latches are loaded in and the cycle repeats.

The three serial dot lines connect to RIN, GIN, and BIN of U337, the VIDATTR PAL. Other inputs to U337 include the FLASH line and three enable lines at pins 1, 2, and 3.

## CIRCUIT DESCRIPTION

---

### Video Logic Circuit Board

When they are asserted, the enable lines from the PIA (U345) gate their respective dot video color to the outputs at pins 14, 15, and 16.

When the  $\overline{\text{FLASH}}$  line – also from the PIA – is asserted, the output lines selected by the enable lines will go high, saturating that color onto the screen and masking any video data on that line.

For example, if  $\overline{\text{ENBL-G}}$  were the only asserted enable line, then dot video would only be present on GOUT. Asserting the  $\overline{\text{FLASH}}$  line would cause GOUT to go to logic one, causing the screen to appear solid green.

Two other lines enter U337; the display enable and the cursor signal. The display enable (DISEN) goes low to blank the video data during horizontal and vertical retrace. It comes from pin 18 of the CRT-C and is delayed by two character clocks through the hex D flip-flop. This delay is used to match the timing of DISEN to the video signal delayed by the parallel-in/serial-out converters. If DISEN wasn't delayed, retrace blanking will occur two clock cycles early, blanking the last two character positions.

The cursor signal enters pin 6 to generate a cursor at ROUT, GOUT, and BOUT. It comes from pin 19 of the CRT-C and goes through the hex D flip-flop to be delayed by two character clocks. This two-character delay places the cursor to the right of the last displayed character.

The horizontal and vertical sync pulses also come from the CRT-C and are clocked through the D flip-flop. These signals, however, bypass U337 and connect to U329, another hex D flip-flop. The RGB lines enter this flip-flop at pins 11, 13, and 14. All five signals are clocked out by the dot clock entering at pin 9. The purpose of this flip-flop is to correct for any propagation delays in the various signal paths.

## CIRCUIT DESCRIPTION

---

### Video Logic Circuit Board

#### Video Output

##### Color Output

The 3 RGB lines from U329 connect to U307. This buffer provides red, green, and blue video pulses to P303. Logic 1 equals color on; 0 is black level.

The horizontal and vertical sync pulses connect to U320, pins 12 and 9. These signals then pass through the drivers at U322 to P303. P303 connects through a mating cable to an RGB color monitor. Jumpers J302 and J304 allow selecting the polarity of the sync signals, while J303 allows sending composite sync to U320 pin 9 by connecting it to U355 pin 11.

##### Monochrome Output

RDOTA, GDOTA, and BDOTA also connect to U323, a 3-to-8-line demultiplexer. J306 and J305 connect one color to each input at pins 1, 2, and 3. If this is a minimum system (green only), pins 1 and 2 are jumpered to pin 3.

U323 decodes the three inputs to assert only one output at Q0-Q7. This signal connects to U309 and is clocked through by the dot clock. The mnemonics on the output lines indicate the color represented by the combination of the three inputs. These outputs couple through the inverters to the resistive weighting network.

This network converts the associated line to a specific voltage level before applying it to the emitter followers at Q302 and Q301. This network forms a monochrome gray scale by controlling the current through the emitter followers.

For example, if RDOTA, GDOTA, and BDOTA were all asserted, U309 pin 19 would go high, driving U308 pin 8 low. This gives the highest resistance in the lower part of the voltage divider, causing the most positive voltage at the output and giving maximum brightness.



## CIRCUIT DESCRIPTION

---

### Video Logic Circuit Board

If none of the RGB lines were asserted U309 pin 2 would go high to place U322 pin 6 at logic zero. This lowers Q301's emitter voltage to the black level.

Composite sync from U355 pin 11 provides horizontal and vertical sync pulses at the blacker-than-black level.

The composite video output of P301 connects to the video input of the internal or external monochrome monitor.

# CIRCUIT DESCRIPTION

---

## Video Logic Circuit Board

### CPU-Video Communications

#### Overview

The CPU can communicate with the video board through several I/O ports, or by read/writing the video RAM. It uses the I/O ports to access the CRT-C, the PIA, and the light pen circuits. It can read/write the video RAM to set up the character font or draw high-density graphics.

#### Video I/O Circuits

Video I/O addresses are decoded by VIOSEL, U369, a 256 × 4 PROM. This IC is selected by the appropriate address on BA0-BA7 and the  $\overline{IO}$  line from the E-clock logic. The outputs are:

$\overline{6845CS}$  Selects the CRT-C programming as described earlier.

$\overline{CRTIOCS}$  (A) Chip-selects the PIA at U345, and (B) provides one input to the OR gate, U372/U366. The other input to this OR gate is  $\overline{6845CS}$ ; the output is  $\overline{4521CS}$ . This line chip-selects U331 when the CPU is reading data from the CRT-C or from the PIA.

$\overline{LPNCS}$  Chip-selects the light-pen counter circuits at U315 if the CPU is processing a light-pen interrupt request. See the discussion on the light-pen circuits.

$\overline{VIDBSEL}$  Asserts when pins 12, 13, or 10 asserts. This line goes to U372 pin 13 and is Nanded with  $\overline{DBIN}$  at U366 pin 13. The result is  $\overline{RDBFRENBL}$  at P304 pin 57; this enables the read buffer, U223, when one of the VIOSEL lines is asserted.

## CIRCUIT DESCRIPTION

---

### Video Logic Circuit Board

Another video I/O circuit is the PIA at U345. This is used for address decoding, controlling the display, and performing some VRAM operations.

The CPU selects the PIA at ECLK time (pin 25) by asserting  $\overline{\text{CRTIOCS}}$  at pin 23. BA0 and BA1 select the register to be accessed while OUT determines if data is to be read from or written to that register. For this PIA, all I/O lines are programmed to be outputs. Here's what they do:

$\overline{\text{ENBL-R}}$ ,  $\overline{\text{-G}}$ ,  $\overline{\text{-B}}$ , &  $\overline{\text{FLASH}}$  Enables the selected video line without affecting RAM.  $\overline{\text{FLASH}}$  causes the selected line to appear as a solid color. See "Converting RAM Data to Video" (Page 4.53) for more information.

$\overline{\text{WRT-R}}$ ,  $\overline{\text{WRT-G}}$ ,  $\overline{\text{WRT-B}}$  Provides a simultaneous write function. When the CPU writes to one color of VRAM, either or both of the other colors may be written into by activating (0) the appropriate line(s) [ $\overline{\text{WRT-R}}$ ,  $\overline{\text{WRT-G}}$ ,  $\overline{\text{WRT-B}}$ ].

$\overline{\text{CRTRAM ENBL}}$  Chip-selects  $\overline{\text{VRAMSEL}}$ , U371 pin 4, which selects the red, green, or blue banks when the CRT reads the VRAM.

$\overline{\text{LA8-LA15}}$  Goes to the memory mapping module to decode the selected video memory location.

$\overline{\text{CLRSCRN}}$  Goes to the video memory circuits to provide a quick means to clear the screen.

# CIRCUIT DESCRIPTION

---

## Video Logic Circuit Board

### Memory Select Circuits

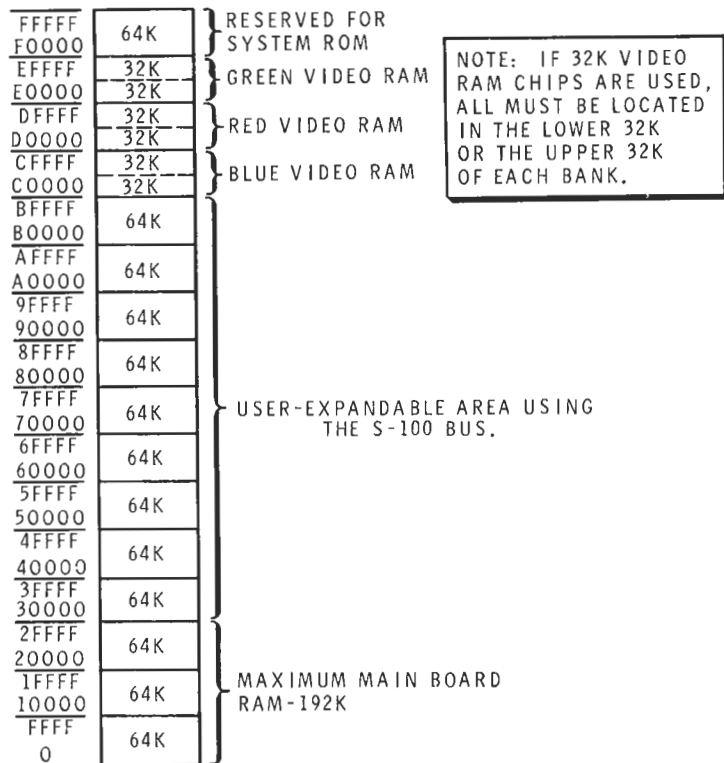
The memory select circuits are centered around U371, VRAMSEL, a  $256 \times 4$  PROM. This IC is used when the CPU wants to access the red, green, or blue memory banks. VRAMSEL is selected when  $\overline{CRTRAM\ ENABLE}$  is asserted at the PIA. Also, MEMR or  $\overline{W\bar{O}}$  is gated through U377 (near VRAMSEL) for further chip-selecting. The OUT line at U377 pin 2 ensures that U371 will not activate on an OUT port operation.

The outputs assert depending on what location in the video memory map is selected. (See Pictorial 4-11.)

RSEL = 0D0000H–0DFFFFH

GSEL = 0E0000H–0EFFFFH

BSEL = 0C0000H–0CFFFFH



**PICTORIAL 4-11**  
Memory Map

## CIRCUIT DESCRIPTION

---

### Video Logic Circuit Board

Video logic boards can have either 32K or 64K parts installed. Current software, however, requires only 32K parts.

$\overline{\text{CRTRAMSEL}}$  asserts whenever pin 11, 10, or 9 asserts. This connects to U372 pin 12 in the lower left corner of the schematic. It is combined with  $\overline{\text{VIDBDSEL}}$  and  $\overline{\text{DBIN}}$  to assert  $\overline{\text{RDBFR ENBL}}$ . This line enables U223 during a memory read operation.

$\overline{\text{CRTRAMSEL}}$  is double-inverted at U366 pin 5 to form  $\overline{\text{CRTRAMSEL1}}$  at P305 pin 61. This line asserts PHANTOM\* at U194 pin 4 on MB2. If an S-100 memory card is occupying the same memory space as VRAM, PHANTOM\* prevents the CPU from writing to the S-100 memory when it is accessing video RAM. This permits you to install read/write memory in the same address space as VRAM without them interfering with each other.

$\overline{\text{CRTRAMSEL}}$  also goes to U372 pin 3,  $\overline{\text{VIDRAMRDY}}$ , through an inverter. If the CRT-C is busy processing a video signal, it will not let the CPU access the RAM circuits. U372 pin 2 is also high, causing  $\overline{\text{VIDRAMRDY}}$  to go low. This drives RDY low at U194 pin 12, putting the CPU into a wait state. The CPU will hold  $\overline{\text{CRTRAMSEL}}$  asserted until the CRT-C gives the CPU control of the video circuits.

Finally,  $\overline{\text{CRTRAMSEL}}$  goes to U379 pin 11, part of the CPU/video arbitration circuits. These circuits synchronize the video circuits to the CPU circuits and determine when the CPU can access the video RAM. See the previous paragraph and the description of the control and timing circuits.

## CIRCUIT DESCRIPTION

### Video Logic Circuit Board

---

#### Read Data Buffers

The CPU reads the addressed data through either U339, U310, or U316. When the CPU reads VRAM, the memory places data on the inputs of these latches. To read a particular bank, the CPU asserts  $\overline{RSEL}$ ,  $\overline{GSEL}$ , or  $\overline{BSEL}$ . For example, to read the data in the green video memory bank, the CPU addresses the desired video memory section (to be explained shortly) and asserts  $\overline{GSEL}$  at U371 pin 10. This signal connects to U351 pin 9. When  $\overline{DBIN}$  from the S-100 bus asserts, U351 pin 8 goes low to couple the data in U310's latches to the BDI bus. In turn, this data couples through U223 to S-100 lines DI0-DI7 before coupling to the CPU.

#### Memory Mapping Module

The memory mapping module consists of U370, U364, and U365. It translates the CPU address range into the address range used by the CRT-C. The CRT-C sees the VRAM in the range of 0-64K, while the CPU sees the memory in the range of 768K to 960K.

To convert the CRT address range to 0-64K, the CPU latches a bit pattern into LA8-LA15. The CPU then requests access of the video RAM by asserting VIDRAMSEL, the desired color bank ( $\overline{RSEL}$ ,  $\overline{BSEL}$ ,  $\overline{GSEL}$ ), and the appropriate address lines on the inputs of U370.

U370 decodes the address and feeds it to the adders at U364 and U365. These ICs add the decoded address to LA8-LA15 and place the result onto the B inputs of U358 and U359. The rest of the CPU address is present on the B inputs of U357 and U363.

When the CRT-C is finished accessing the display, it brings VIDRAMSEL low at U377 pin 4 (lower left corner of schematic). This couples the B inputs of the multiplexer ICs onto address lines DA0-DA15. The correct VRAM location can now be read or written.

## CIRCUIT DESCRIPTION

---

### Video Logic Circuit Board

#### Video RAM

##### Overview

The video RAMs are 32K or 64K × 1-bit dynamic RAMs. The RAMs are arranged into three banks, 64K apart; one bank for each of the primary colors. In a minimum system, only the green bank will contain memory. The CPU can read/write RAM, while the CRT-C can only read.

##### CPU Write

The CPU writes to RAM through U346; it places data onto the bus and asserts WE of each chip through U374 pin 11. This comes from BMWRT and VIDRAMSEL at U355 pin 5 and U351 pin 4.

The RAS portion of the address is present on VA0-VA7.

U350 gates the RAS line through U375 pin 11, U375 pin 8, and U374 pin 8 for the selected bank.

Next, the CAS address is placed on VA0-VA7 and the CAS line asserts U375 pin 3, U375 pin 6, and U374 pin 6. Only the bank(s) previously selected by RAS will be affected.

Data present at the inputs of U346 are coupled into the appropriate memory location(s) in the video RAM.

##### CPU Read

When the CPU reads from RAM, it asserts  $\overline{R-SEL}$ ,  $\overline{G-SEL}$ , or  $\overline{B-SEL}$  to select the appropriate color bank. The RAS and CAS lines operate as before. The address data is placed on the DOUT lines of the selected banks and read by the CPU as explained previously.

# CIRCUIT DESCRIPTION

---

## Video Logic Circuit Board

### CRT-C Read

The CRT-C reads all three banks at the same time; the enable lines at U337 select which banks are to be displayed as explained previously. When the CRT-C has control, VIDRAM-SEL is low and couples to pins 13, 5, and 11 of U350. This forces pins 12, 6, and 8 of U350 to logic 1.

When RAS occurs, the address on VA0-VA7 is latched into all three banks. Next, CAS asserts and also addresses all three banks. The data from each bank is placed onto the appropriate bus and sent to the parallel/serial conversion circuits.

### Clear Screen Function

The clear screen function allows the CPU to quickly clear the screen. Instead of directly writing to memory, which is time-consuming, the CPU uses the fast scanning feature of the CRT-C. Here's how.

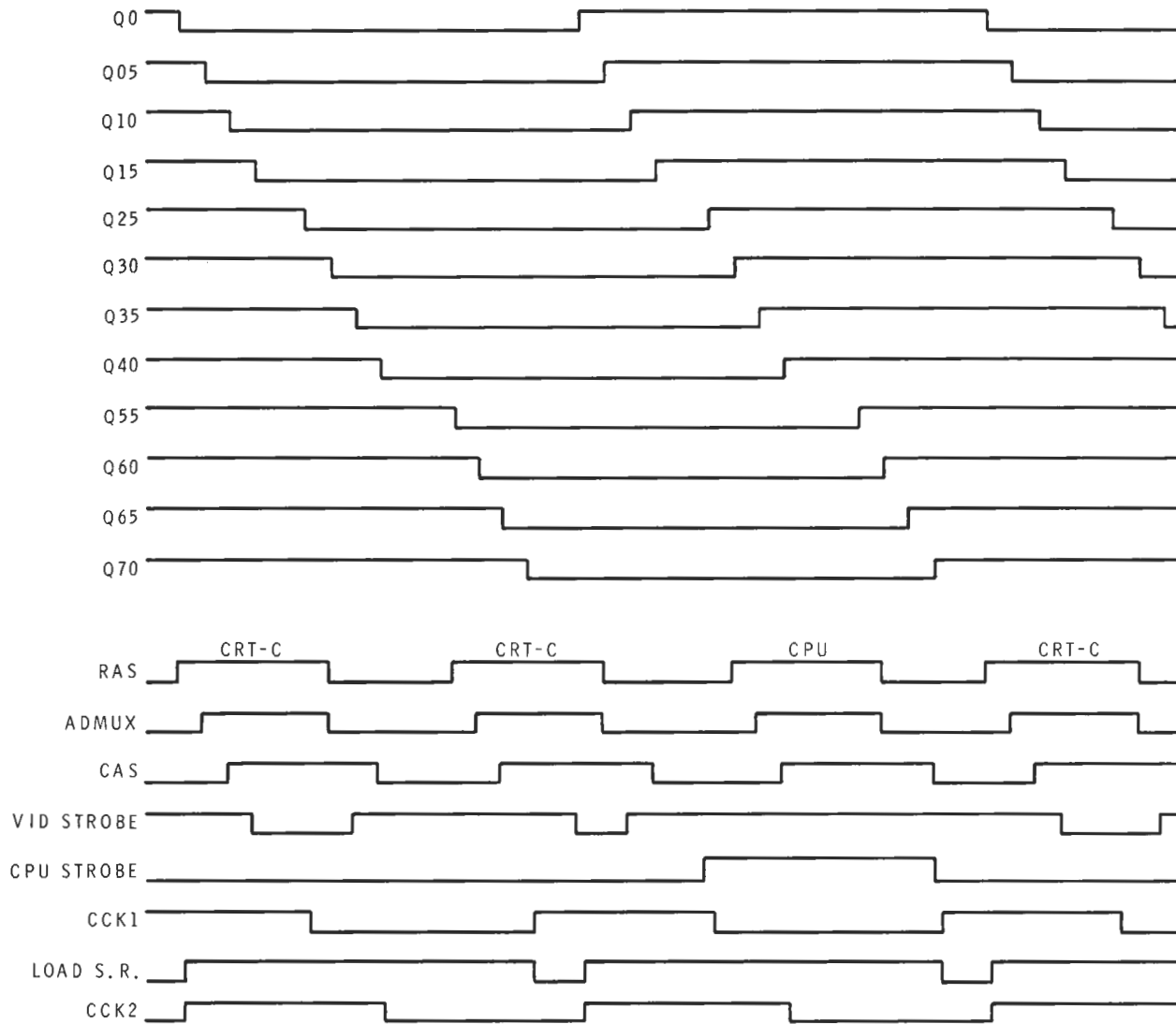
The CPU asserts the CLRSCRN line at the PIA; it also clears or sets the SET line. These lines connect to the PAL at U346; CLRSCRN disconnects the PAL from the data on its input, while SET places all ones or zeros on the output lines, depending on the logic level at pin 11. (If the level is logic one, the screen will be painted white instead of blanked.)

CLRSCRN also connects to U355 pin 4 and U351 pin 5 to force the  $\overline{WE}$  line low on all RAMs. During this time, the CRT-C has control of the bus. Since the CRT-C scans all memory locations, each bank will be filled with ones or zeros, depending on the level on SET. The CRT is quickly blanked or flashes white.



# CIRCUIT DESCRIPTION

## Video Logic Circuit Board



**Pictorial 4-12**  
Video Board Timing

## CIRCUIT DESCRIPTION

### Video Logic Circuit Board

#### Timing and Video Arbitration

##### Timing

Refer to the Schematic Diagram and Pictorial 4-12 as you read the following materials.

The 14.112 MHz crystal-controlled oscillator at U368 provides the basic timing for the video circuits. This signal couples through U344 pin 11 to provide dot clock and couples through U344 pin 6 for inverted dot clock. This method was used instead of series-connected inverters to ensure that the two clock signals are exactly 180-degrees out of phase.

DOTCLK drives U336 and U343; these ICs are wired as a ring counter to derive Q0-Q70 shown in the adjacent waveforms. U367, driven by  $\overline{\text{DOTCLK}}$ , uses some of these outputs to generate the odd-numbered waveforms from Q05 to Q65. These signals connect to the VIDRAM PAL at U376.

U376 uses the Q signals to generate VIDSTRB, ADMUX, RAS, and CAS. VIDSTRB clocks addressed data into the latches prior to parallel-to-serial conversion as described previously. ADMUX multiplexes the 16-bit address bus onto an 8-bit address bus in time with RAS and CAS. ADMUX is low during RAS and high during CAS.

The CRT-C has control of the video circuits for 2/3 of any timing cycle. This ensures fast display refresh while the remaining 1/3 allows the CPU to rapidly update the display memory.

The CRT-C's portion of the cycle begins on the negative transition of Q0. This is indicated by the two RAS waveforms marked "CRT-C" on the Video Board Timing waveforms. Video arbitration circuits (to be explained presently) ensure that the CRT-C always has control during these two RAS cycles.

## CIRCUIT DESCRIPTION

---

### Video Logic Circuit Board

The third RAS cycle of the video timing cycle is reserved for the CPU. If the CPU doesn't attempt to read or write memory, RAS will not assert during the time marked "CPU." If the CPU does attempt to read or write memory RAS will assert and the memory access can take place. Note that during CPU RAS time, VIDSTRB (U376 pin 17) does not pulse. This prevents the addressed memory location from being latched into U332, U302, and U311; keeping unwanted noise off the display.

If the CPU attempts to access video memory during the CRT-C portion of the cycle, the arbitration circuits places a logic zero on P305 pin 62. This logic zero couples to the CPU's READY line which puts the CPU into a wait state. The CPU ceases activity until the "CPU" RAS cycle begins. At this time, P305 pin 62 goes high to activate the CPU.

Obviously, the CPU processing time will slow down if it performs a lot of reading and writing to video RAM. However, the video arbitration circuits do not slow down the CPU for non-video operations (such as I/O and system memory accesses). As long as the CPU isn't accessing the video circuits, P305 pin 62 remains high and the CPU operates at full speed.

Now for a closer look at the video arbitration circuits.

#### Video Arbitration

The video arbitration circuits determine if the CPU is requesting access to the video RAM. If the CRT-C is not using the RAM, it gives control to the CPU. However, the CRT-C always has priority.

As mentioned previously, the CPU requests control of the VRAM by asserting  $\overline{\text{RSEL}}$ ,  $\overline{\text{GSEL}}$ , or  $\overline{\text{BSEL}}$  at U371. This asserts  $\overline{\text{CRTRAMSEL}}$  which couples through U372 pin 3 to put the CPU into a wait state after the CPU finishes the 2nd processor cycle.

## CIRCUIT DESCRIPTION

### Video Logic Circuit Board

$\overline{\text{CRTRAMSEL}}$  also goes to U379 pin 11 to set up the bus arbitration circuits for a read/write request from the CPU. If the operation is a CPU write, then U379 pin 3 goes high. If the operation is a CPU read, then MEMR is clocked into U378 pin 5 when  $\text{STVAL}^*\text{SYNC}$  asserts. In turn, U378 pin 5 couples the logic one to U379 pin 2.

At this time, U361 pin 8 is latched to logic one which is coupled to U372 pin 2. U372 pin 1 is also logic one due to the asserted  $\overline{\text{CRTRAMSEL}}$  line at U366 pin 4. U372 pin 3 holds the CPU in a wait state as described previously. Because of this, pins 11 and 12 of U379 remain at logic zero. The resulting logic 1 at U379 pin 13 is the CPU request signal which couples to pin 2 of U361.

When the CRT-C has completed processing the video circuits, Q15 at U361 pin 3 goes high. This latches U361 pin 6 to logic zero and, because U361 pin 9 is also zero, drives the  $\overline{\text{VIDRAMSEL}}$  line at U377 pin 4 to logic one.  $\overline{\text{VIDRAMSEL}}$  connects to the control inputs of the CPU/CRT-C address multiplexers to couple the CPU address lines to the video memory circuits.

If the CPU is writing memory, data from the S-100 bus is present on lines BD00-BD07. BMWRT writes this data into memory. If the CPU is reading memory, the address memory location places the data onto U339, U310, or U316; depending on the RGB select lines going into memory.

When line Q65 goes high, the logic one at U361 pin 5 is latched into U361 pin 9. This latches the data on the inputs of U339, U310, and U316 onto their outputs; if memory read. The status of the gate at the input of each octal latch will determine which latch will be coupled to the bus.

## CIRCUIT DESCRIPTION

---

### Video Logic Circuit Board

At the same time, U361 pin 8 goes low to bring  $\overline{\text{VIDRAMRDY}}$  high. The CPU leaves the wait state and finishes processing the instruction.  $\overline{\text{CRTRAMSEL}}$  goes high to drive U379 pin 13 low. Since VIDRAMSEL is also low, U355 pin 3 goes to logic zero to clear U361.

The CRT-C again has control of the video board.

### Light Pen Circuits

The light pen strobe enters U362 pin 12 from U116 pin 9 (see the parallel port description for more detail). The  $\overline{\text{DOTCLK}}$  signal toggles LTPNSTB through U362 pin 9 to U356 pin 5. Next, the clock signal at U356 pin 11 latches the LTPNSTB signal onto U356 pin 9. This positive-going signal latches the refresh address into the CRT-C's light pen register. See the CRT-C IC data sheets.

Also, the output of U356 pin 5, PENSTBD, goes to U315 pin 11. U315 is an octal latch that is loaded by the CRT-C row address lines, RA0-RA3, and the 4-bit down-counter, U324.

At the time of PENSTBD, RA0-RA3 point to the row that was active when the light pen strobe occurred; U324 points to the dot position.

As explained in the parallel port description, when LTPNSTB asserts, the parallel port sends an interrupt to the CPU. From here, it is up to the user's program to process the interrupt.

If the CPU is programmed to respond to a light-pen interrupt, it will read the data stored in the CRT-C light-pen register and the data stored U315 to find the exact pixel location. The CPU reads the CRT-C as described earlier; it reads U315 by asserting  $\overline{\text{LTPNCS}}$  from the VIOSEL PROM and  $\overline{\text{DBIN}}$  from the S-100 bus. From here, the CPU can compute the video memory location and access the bit in that memory location to be processed.

## TROUBLESHOOTING

---

Use the following chart for help in identifying the source of problems. The chart lists conditions and possible causes for specific problems. If you cannot resolve the problem, refer to the warranty and service information supplied with your Computer.

You may wish to service some problems yourself. In the following chart, if a particular part is mentioned, check that part and other components that are associated with it. Remember to locate and correct the cause when components are damaged, or the problem could reoccur.

Refer to the "Circuit Board X-Ray View" for the physical location of parts on the circuit boards.

CONDITON	POSSIBLE CAUSE
Monitor blank	<ol style="list-style-type: none"> <li>1. Not plugged in.</li> <li>2. Not turned on.</li> <li>3. Cables P304 or P305 not connected properly.</li> <li>4. Power supply.</li> </ol>
Vertical roll	<ol style="list-style-type: none"> <li>1. Jumper 301 in wrong position.</li> </ol>
Horizontal tear	<ol style="list-style-type: none"> <li>1. Jumper 302 in wrong position.</li> </ol>
Random dots	<ol style="list-style-type: none"> <li>1. Jumper 307 in wrong position.</li> </ol>
Dark screen	<ol style="list-style-type: none"> <li>1. Adjust R307 (Black Level control).</li> </ol>
Vertical lines filling the entire usable video screen.	<ol style="list-style-type: none"> <li>1. One or more Z-219-1 video RAM ICs installed backwards.</li> </ol>

## REPLACEMENT PARTS LIST

## Video Logic Circuit Board

CIRCUIT Comp. No.	HEATH Part No.	Description
----------------------	-------------------	-------------

**Resistors**

All resistors are 1/4-watt, 5%, unless specified otherwise.

R301	10-1204	1000 $\Omega$ control (may not be in all units)
RP301	9-99	1000 $\Omega$ resistor pack
RP302	9-128	10 k $\Omega$ resistor pack
R303	6-102-12	1000 $\Omega$
RP303	9-124	4700 $\Omega$ resistor pack
R304	6-470-12	47 $\Omega$
RP304	9-124	4700 $\Omega$ resistor pack
R305	6-102-12	1000 $\Omega$
RP305	9-93	33 $\Omega$ resistor pack
R306	6-270-12	27 $\Omega$
RP306	9-93	33 $\Omega$ resistor pack
R307	10-1191	100 $\Omega$ control
RP307	9-99	1000 $\Omega$ resistor pack
R308	6-621-12	620 $\Omega$
R309	6-221-12	220 $\Omega$
R310	6-111-12	110 $\Omega$
R311	6-330-12	33 $\Omega$
R312	6-470-12	47 $\Omega$
R313	6-270-12	27 $\Omega$
R314	6-390-12	39 $\Omega$
R315	6-620-12	62 $\Omega$
R316	6-650-12	27 $\Omega$
R317	6-102-12	1000 $\Omega$
R318	6-102-12	1000 $\Omega$
R319	6-103-12	10 k $\Omega$
R320	6-103-12	10 k $\Omega$
R321	6-102-12	1000 $\Omega$
R322	6-472-12	4700 $\Omega$
R325	6-102-12	1000 $\Omega$

# REPLACEMENT PARTS LIST

---

## Video Logic Circuit Board

<u>CIRCUIT</u>	<u>HEATH</u>	<u>DESCRIPTION</u>
<u>Comp. No.</u>	<u>Part No.</u>	

### Capacitors

C301-C302	21-746	180 pF ceramic
C303	21-762	.1 $\mu$ F ceramic
C304	25-820	10 $\mu$ F electrolytic
C305-C307	21-762	.1 $\mu$ F ceramic
C308	25-820	10 $\mu$ F electrolytic
C309-C335	21-762	.1 $\mu$ F ceramic
C336	25-820	10 $\mu$ F electrolytic
C337-C363	21-762	.1 $\mu$ F ceramic
C364	25-883	47 $\mu$ F electrolytic
C365	21-762	.1 $\mu$ F ceramic
C366-C368	21-746	180 pF ceramic

### Miscellaneous

L301	475-15	1.22 $\mu$ H ferrite bead
L302-L304	235-229	35 mH coil
U368	150-134	14.112 MHz crystal oscillator

### Semiconductors

See the "Semiconductor Identification Chart."



# SEMICONDUCTOR IDENTIFICATION

---

## Component Number Index

This section is divided into four parts. The "Component Number Index" relates circuit component numbers to Heath Part Numbers. The "Part Number Index" relates part numbers to manufacturers' part numbers, as well as providing lead configuration drawings for each part. The remaining two parts are "PAL Equations" and "ROM Codes" for the PALs and ROMs on the video logic circuit board.

CIRCUIT COMPONENT NUMBER	HEATH PART NUMBER
Q301	417-118
Q302	417-118
U301	443-892
U302	443-805
U303	443-892
U304	443-1106*
U305	443-1106*
U306	443-1106*
U307	443-791
U308	443-967
U309	443-805
U310	443-837
U311	443-837
U312	443-1106*
U313	443-1106*
U314	443-1106*
U315	443-863
U316	443-837
U317	443-1106*
U318	443-1106*
U319	443-1106*
U320	443-891
U321	443-879
U322	443-967
U323	443-804
U324	443-1054
U325	443-892
U326	443-1106*
U327	443-1106*
U328	443-1106*
U329	443-1053
U330	443-1013
U331	443-1058
U332	443-805
U333	443-1106*
U334	443-1106*
U335	443-1106*

These IC's may be 443-1106 32K × 1 RAM ICs or 443-970 64K × 1 RAM ICs.

# SEMICONDUCTOR IDENTIFICATION

## Component Number Index

CIRCUIT COMPONENT NUMBER	HEATH PART NUMBER
U336	443-983
U337	443-115
U338	443-1058
U339	443-837
U340	443-1106*
U341	443-1106*
U342	443-1106*
U343	443-983
U344	443-915
U345	443-1014
U346	444-133
U347	443-1106*
U348	443-1106*
U349	443-1106*
U350	443-797
U351	443-875
U352	443-1106*
U353	443-1106*
U354	443-1106*
U355	443-875
U356	443-1051
U357	443-799
U358	443-799
U359	443-799
U360	443-1057
U361	443-1051
U362	443-1051
U363	443-799
U364	443-855
U365	443-855
U366	443-754
U367	443-1053
U369	443-103
U370	443-127
U371	443-102
U372	443-1049
U373	443-1057
U374	443-1049
U375	443-1049
U376	444-114
U377	443-1048
U378	443-1051
U379	443-1045

These IC's may be 443-1106 32K × 1 RAM ICs or 443-970 64K × 1 RAM ICs.

# SEMICONDUCTOR IDENTIFICATION

## Part Number Index

This index shows a lead configuration detail (basing diagram) of each semiconductor part number.

### Transistors

HEATH PART NUMBER	MAY BE REPLACED WITH	DESCRIPTION	LEAD CONFIGURATION
417-118	2N3393	NPN	

### Integrated Circuits

HEATH PART NUMBER	MAY BE REPLACED WITH	DESCRIPTION	LEAD CONFIGURATION
443-754	74LS240	Tri-state octal buffer	
443-791	74LS244	Tri-state buffer driver	

(cont'd)

# SEMICONDUCTOR IDENTIFICATION

## Part Number Index

### Integrated Circuits (cont'd)

HEATH PART NUMBER	MAY BE REPLACED WITH	DESCRIPTION	LEAD CONFIGURATION
443-797	74LS10	Triple 3-input NAND	
443-799	74LS157	Quad 2-line-to-1-line Multipliers	
443-804	74LS259	8-bit latch	

(cont'd)

# SEMICONDUCTOR IDENTIFICATION

## Part Number Index

Integrated Circuits (cont'd)

HEATH PART NUMBER	MAY BE REPLACED WITH	DESCRIPTION	LEAD CONFIGURATION
443-805	74LS273	Octal D flip-flop	
443-837	74LS373	Octal D latch	
443-855	74LS283	Adder	
443-863	74LS374	Octal D flip-flop	

(cont'd)

# SEMICONDUCTOR IDENTIFICATION

## Part Number Index

### Integrated Circuits (cont'd)

HEATH PART NUMBER	MAY BE REPLACED WITH	DESCRIPTION	LEAD CONFIGURATION
443-875	74LS32	Quad 2 input OR	
443-879	74LS174	Hex D flip-flop	
443-891	74LS86	Quad 2-input Exclusive-OR	
443-892	74LS166	Parallel In Serial Out Shift Register	

(cont'd)

# SEMICONDUCTOR IDENTIFICATION

## Part Number Index

### Integrated Circuits (cont'd)

HEATH PART NUMBER	MAY BE REPLACED WITH	DESCRIPTION	LEAD CONFIGURATION
443-915	74S86	Quad 2-input Exclusive-OR	
443-967	7406	Hex inverter	
443-983	74S175	Quad D flip-flop	
443-1013	68A45	CRT controller	

(cont'd)

# SEMICONDUCTOR IDENTIFICATION

## Part Number Index

### Integrated Circuits (cont'd)

HEATH PART NUMBER	MAY BE REPLACED WITH	DESCRIPTION	LEAD CONFIGURATION
443-1014	68A21	PIA	
443-1045	74ALS02	Quad 2-input NOR	
443-1048	74ALS28	Quad buffer NOR	
443-1049	74ALS37	NAND buffer	
443-1051	74ALS74	Dual D flip-flop	

(cont'd)



# SEMICONDUCTOR IDENTIFICATION

## Part Number Index

### Integrated Circuits (cont'd)

HEATH PART NUMBER	MAY BE REPLACED WITH	DESCRIPTION	LEAD CONFIGURATION
443-1053	74S174	Hex D flip-flop	
443-1054	74LS169	Up down counter	
443-1057	74S241	Octal buffer	
443-1058	74LS541	Octal buffer	

(cont'd)

## SEMICONDUCTOR IDENTIFICATION

## Part Number Index

## Integrated Circuits (cont'd)

HEATH PART NUMBER	MAY BE REPLACED WITH	DESCRIPTION	LEAD CONFIGURATION
443-1106	MCM66330	RAM 32k × 1	
443-970	MCM6665	RAM 64K × 1	
444-102	Available only from Zenith Data Systems or Heath Company	Video memory decoder	
444-103	Available only from Zenith Data Systems or Heath Company	Video I/O decoder	
444-114	PAL or HAL14L4 Available only from Zenith Data Systems or Heath Company	Video control	
444-115	PAL or HAL14H4 Available only from Zenith Data Systems or Heath Company	RAM control	

(cont'd)

# SEMICONDUCTOR IDENTIFICATION

## Part Number Index

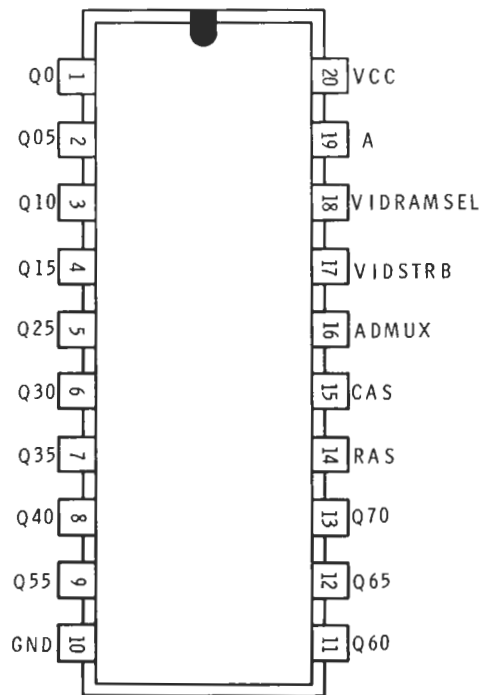
### Integrated Circuits (cont'd)

HEATH PART NUMBER	MAY BE REPLACED WITH	DESCRIPTION	LEAD CONFIGURATION
444-127	Available only from Zenith Data Systems or Heath Company 18S22	PROM	
444-133	Available only from Zenith Data Systems or Heath Company HAL10H8 or PAL	Video clear screen	

# SEMICONDUCTOR IDENTIFICATION

## PAL Equations

### 444-114/Video Ram Controller



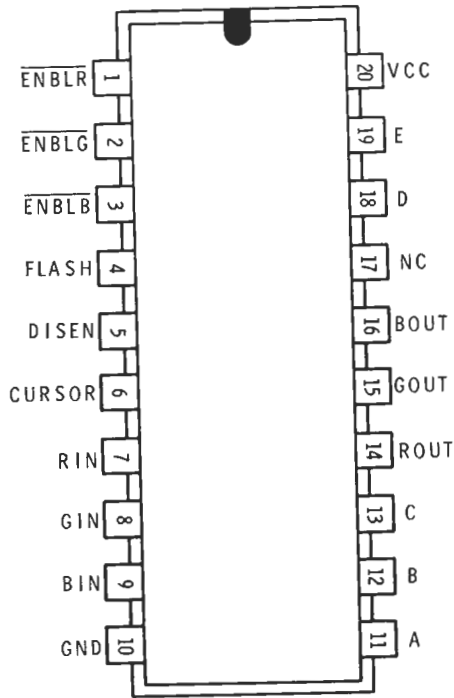
### LOGIC EQUATIONS

$$\begin{aligned}
 \text{RAS} &= \overline{Q30} \cdot Q55 + Q05 \cdot \overline{Q30} + Q0 \cdot Q60 + Q25 \cdot \overline{Q70} \cdot \overline{\text{VIDRAMSEL}} \\
 \text{CAS} &= \overline{Q40} \cdot Q65 + Q15 \cdot \overline{Q40} + Q70 \cdot Q10 + Q0 \cdot \overline{Q30} \cdot \overline{\text{VIDRAMSEL}} \\
 \text{ADMUX} &= \overline{Q05} \cdot Q30 + \overline{Q60} \cdot \overline{Q05} + Q35 \cdot Q60 \cdot \overline{\text{VIDRAMSEL}} \\
 \text{VIDSTRB} &= \overline{Q15} \cdot Q35 + Q0 \cdot \overline{Q10}
 \end{aligned}$$

# SEMICONDUCTOR IDENTIFICATION

## PAL Equations

### 444-115/Video Attribute Controller



### LOGIC EQUATIONS

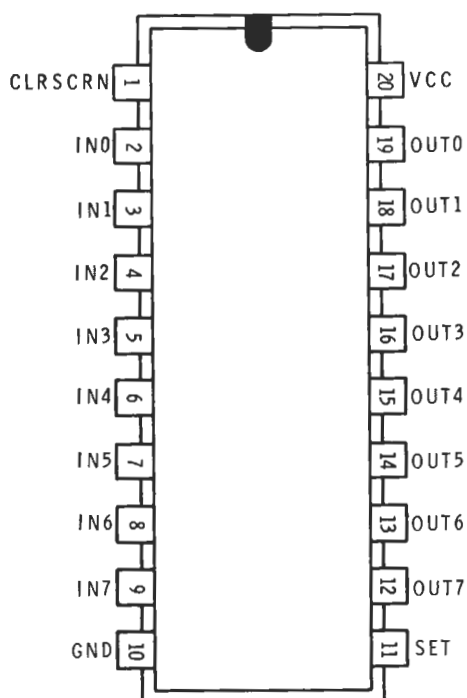
$$\begin{aligned}
 \text{ROUT} &= \text{DISEN} * \text{ENBLR} * \overline{\text{FLASH}} + \text{DISEN} * \text{ENBLR} * \text{CURSOR} * \overline{\text{RIN}} \\
 &\quad + \text{DISEN} * \text{ENBLR} * \overline{\text{CURSOR}} * \text{RIN} \\
 \text{GOUT} &= \text{DISEN} * \text{ENBLG} * \overline{\text{FLASH}} + \text{DISEN} * \text{ENBLG} * \text{CURSOR} * \overline{\text{GIN}} \\
 &\quad + \text{DISEN} * \text{ENBLG} * \overline{\text{CURSOR}} * \text{GIN} \\
 \text{ROUT} &= \text{DISEN} * \text{ENBLB} * \overline{\text{FLASH}} + \text{DISEN} * \text{ENBLB} * \text{CURSOR} * \overline{\text{BIN}} \\
 &\quad + \text{DISEN} * \text{ENBLB} * \overline{\text{CURSOR}} * \text{BIN}
 \end{aligned}$$

# SEMICONDUCTOR IDENTIFICATION

## PAL Equations

### 444-133/CLRSCRN

Data Buffer for Video RAM with  
Clear Screen and Set Screen Functions



#### LOGIC EQUATIONS

$$\begin{aligned} \text{OUT0} &= \overline{\text{CLRSCRN}} * \text{IN0} + \text{CLRSCRN} * \text{SET} \\ \text{OUT1} &= \overline{\text{CLRSCRN}} * \text{IN1} + \text{CLRSCRN} * \text{SET} \\ \text{OUT2} &= \overline{\text{CLRSCRN}} * \text{IN2} + \text{CLRSCRN} * \text{SET} \\ \text{OUT3} &= \overline{\text{CLRSCRN}} * \text{IN3} + \text{CLRSCRN} * \text{SET} \\ \text{OUT4} &= \overline{\text{CLRSCRN}} * \text{IN4} + \text{CLRSCRN} * \text{SET} \\ \text{OUT5} &= \overline{\text{CLRSCRN}} * \text{IN5} + \text{CLRSCRN} * \text{SET} \\ \text{OUT6} &= \overline{\text{CLRSCRN}} * \text{IN6} + \text{CLRSCRN} * \text{SET} \\ \text{OUT7} &= \overline{\text{CLRSCRN}} * \text{IN7} + \text{CLRSCRN} * \text{SET} \end{aligned}$$

## SEMICONDUCTOR IDENTIFICATION

## ROM Codes

```

                                title   VRAMSEL video ram select prom for the Z-100
;                                ZDS part no.: 444-102
;                                release date: 5/21/82
;                                prom:      32s129 (256x4)
;                                checksum:  0eef

000C      red_en equ    0ch
0006      blu_en equ    06h
000A      grn_en equ    0ah

0000'
                                cseg
                                org      0

0000'    0F                db      0fh
0001'    0F                db      0fh
0002'    0F                db      0fh
0003'    0F                db      0fh
0004'    0F                db      0fh
0005'    0F                db      0fh
0006'    0F                db      0fh
0007'    0F                db      0fh
0008'    0F                db      0fh
0009'    0F                db      0fh
000A'    0F                db      0fh
000B'    0F                db      0fh
000C'    06                db      blu_en
000D'    0C                db      red_en
000E'    0A                db      grn_en
000F'    0F                db      0fh
0010'    0F                db      0fh
0011'    0F                db      0fh
0012'    0F                db      0fh
0013'    0F                db      0fh
0014'    0F                db      0fh
0015'    0F                db      0fh
0016'    0F                db      0fh
0017'    0F                db      0fh
0018'    0F                db      0fh
0019'    0F                db      0fh
001A'    0F                db      0fh
001B'    0F                db      0fh
001C'    0F                db      0fh
001D'    0F                db      0fh
001E'    0F                db      0fh
001F'    0F                db      0fh

```

## SEMICONDUCTOR IDENTIFICATION

## ROM Codes

VRAMSEL video ram select prom for the Z-100

0020'	0F	db	0fh
0021'	0F	db	0fh
0022'	0F	db	0fh
0023'	0F	db	0fh
0024'	0F	db	0fh
0025'	0F	db	0fh
0026'	0F	db	0fh
0027'	0F	db	0fh
0028'	0F	db	0fh
0029'	0F	db	0fh
002A'	0F	db	0fh
002B'	0F	db	0fh
002C'	0F	db	0fh
002D'	0F	db	0fh
002E'	0F	db	0fh
002F'	0F	db	0fh
0030'	0F	db	0fh
0031'	0F	db	0fh
0032'	0F	db	0fh
0033'	0F	db	0fh
0034'	0F	db	0fh
0035'	0F	db	0fh
0036'	0F	db	0fh
0037'	0F	db	0fh
0038'	0F	db	0fh
0039'	0F	db	0fh
003A'	0F	db	0fh
003B'	0F	db	0fh
003C'	0F	db	0fh
003D'	0F	db	0fh
003E'	0F	db	0fh
003F'	0F	db	0fh
0040'	0F	db	0fh
0041'	0F	db	0fh
0042'	0F	db	0fh
0043'	0F	db	0fh
0044'	0F	db	0fh
0045'	0F	db	0fh
0046'	0F	db	0fh
0047'	0F	db	0fh
0048'	0F	db	0fh
0049'	0F	db	0fh
004A'	0F	db	0fh
004B'	0F	db	0fh
004C'	0F	db	0fh
004D'	0F	db	0fh



## SEMICONDUCTOR IDENTIFICATION

## ROM Codes

VRAMSEL video ram select prom for the Z-100

004E'	0F	db	0fh
004F'	0F	db	0fh
0050'	0F	db	0fh
0051'	0F	db	0fh
0052'	0F	db	0fh
0053'	0F	db	0fh
0054'	0F	db	0fh
0055'	0F	db	0fh
0056'	0F	db	0fh
0057'	0F	db	0fh
0058'	0F	db	0fh
0059'	0F	db	0fh
005A'	0F	db	0fh
005B'	0F	db	0fh
005C'	0F	db	0fh
005D'	0F	db	0fh
005E'	0F	db	0fh
005F'	0F	db	0fh
0060'	0F	db	0fh
0061'	0F	db	0fh
0062'	0F	db	0fh
0063'	0F	db	0fh
0064'	0F	db	0fh
0065'	0F	db	0fh
0066'	0F	db	0fh
0067'	0F	db	0fh
0068'	0F	db	0fh
0069'	0F	db	0fh
006A'	0F	db	0fh
006B'	0F	db	0fh
006C'	0F	db	0fh
006D'	0F	db	0fh
006E'	0F	db	0fh
006F'	0F	db	0fh
0070'	0F	db	0fh
0071'	0F	db	0fh
0072'	0F	db	0fh
0073'	0F	db	0fh
0074'	0F	db	0fh
0075'	0F	db	0fh
0076'	0F	db	0fh
0077'	0F	db	0fh
0078'	0F	db	0fh
0079'	0F	db	0fh
007A'	0F	db	0fh
007B'	0F	db	0fh

## SEMICONDUCTOR IDENTIFICATION

## ROM Codes

VRAMSEL video ram select prom for the Z-100

007C'	0F	db	0fh
007D'	0F	db	0fh
007E'	0F	db	0fh
007F'	0F	db	0fh
0080'	0F	db	0fh
0081'	0F	db	0fh
0082'	0F	db	0fh
0083'	0F	db	0fh
0084'	0F	db	0fh
0085'	0F	db	0fh
0086'	0F	db	0fh
0087'	0F	db	0fh
0088'	0F	db	0fh
0089'	0F	db	0fh
008A'	0F	db	0fh
008B'	0F	db	0fh
008C'	0F	db	0fh
008D'	0F	db	0fh
008E'	0F	db	0fh
008F'	0F	db	0fh
0090'	0F	db	0fh
0091'	0F	db	0fh
0092'	0F	db	0fh
0093'	0F	db	0fh
0094'	0F	db	0fh
0095'	0F	db	0fh
0096'	0F	db	0fh
0097'	0F	db	0fh
0098'	0F	db	0fh
0099'	0F	db	0fh
009A'	0F	db	0fh
009B'	0F	db	0fh
009C'	0F	db	0fh
009D'	0F	db	0fh
009E'	0F	db	0fh
009F'	0F	db	0fh
00A0'	0F	db	0fh
00A1'	0F	db	0fh
00A2'	0F	db	0fh
00A3'	0F	db	0fh
00A4'	0F	db	0fh
00A5'	0F	db	0fh
00A6'	0F	db	0fh
00A7'	0F	db	0fh
00A8'	0F	db	0fh
00A9'	0F	db	0fh

## SEMICONDUCTOR IDENTIFICATION

## ROM Codes

VRAMSEL video ram select prom for the Z-100

00AA'	0F	db	0fh
00AB'	0F	db	0fh
00AC'	0F	db	0fh
00AD'	0F	db	0fh
00AE'	0F	db	0fh
00AF'	0F	db	0fh
00B0'	0F	db	0fh
00B1'	0F	db	0fh
00B2'	0F	db	0fh
00B3'	0F	db	0fh
00B4'	0F	db	0fh
00B5'	0F	db	0fh
00B6'	0F	db	0fh
00B7'	0F	db	0fh
00B8'	0F	db	0fh
00B9'	0F	db	0fh
00BA'	0F	db	0fh
00BB'	0F	db	0fh
00BC'	0F	db	0fh
00BD'	0F	db	0fh
00BE'	0F	db	0fh
00BF'	0F	db	0fh
00C0'	0F	db	0fh
00C1'	0F	db	0fh
00C2'	0F	db	0fh
00C3'	0F	db	0fh
00C4'	0F	db	0fh
00C5'	0F	db	0fh
00C6'	0F	db	0fh
00C7'	0F	db	0fh
00C8'	0F	db	0fh
00C9'	0F	db	0fh
00CA'	0F	db	0fh
00CB'	0F	db	0fh
00CC'	0F	db	0fh
00CD'	0F	db	0fh
00CE'	0F	db	0fh
00CF'	0F	db	0fh
00D0'	0F	db	0fh
00D1'	0F	db	0fh
00D2'	0F	db	0fh
00D3'	0F	db	0fh
00D4'	0F	db	0fh
00D5'	0F	db	0fh
00D6'	0F	db	0fh
00D7'	0F	db	0fh

## SEMICONDUCTOR IDENTIFICATION

## ROM Codes

VRAMSEL video ram select prom for the Z-100

00D8'	0F	db	0fh
00D9'	0F	db	0fh
00DA'	0F	db	0fh
00DB'	0F	db	0fh
00DC'	0F	db	0fh
00DD'	0F	db	0fh
00DE'	0F	db	0fh
00DF'	0F	db	0fh
00E0'	0F	db	0fh
00E1'	0F	db	0fh
00E2'	0F	db	0fh
00E3'	0F	db	0fh
00E4'	0F	db	0fh
00E5'	0F	db	0fh
00E6'	0F	db	0fh
00E7'	0F	db	0fh
00E8'	0F	db	0fh
00E9'	0F	db	0fh
00EA'	0F	db	0fh
00EB'	0F	db	0fh
00EC'	0F	db	0fh
00ED'	0F	db	0fh
00EE'	0F	db	0fh
00EF'	0F	db	0fh
00F0'	0F	db	0fh
00F1'	0F	db	0fh
00F2'	0F	db	0fh
00F3'	0F	db	0fh
00F4'	0F	db	0fh
00F5'	0F	db	0fh
00F6'	0F	db	0fh
00F7'	0F	db	0fh
00F8'	0F	db	0fh
00F9'	0F	db	0fh
00FA'	0F	db	0fh
00FB'	0F	db	0fh
00FC'	0F	db	0fh
00FD'	0F	db	0fh
00FE'	0F	db	0fh
00FF'	0F	db	0fh
		end	

# SEMICONDUCTOR IDENTIFICATION

---

## ROM Codes

VRAMSEL video ram select prom for the Z-100

Macros:

Symbols:

BLU\_EN 0006    GRN\_EN 000A    RED\_EN 000C

No Fatal error(s)

## SEMICONDUCTOR IDENTIFICATION

## ROM Codes

```

                                title  VIOSEL - video i/o select prom
                                ;
                                ;      ZDS part no.:  444-103
                                ;      release date:  5/21/82
                                ;      prom:         82s129
                                ;      checksum:     0eba
                                ;

0000'                                cseg
                                org      0

0005                                sel6821 equ  0101b
0006                                sel6845 equ  0110b
0003                                lightpen equ  0011b

0000'  0F                                db      00fh
0001'  0F                                db      00fh
0002'  0F                                db      00fh
0003'  0F                                db      00fh
0004'  0F                                db      00fh
0005'  0F                                db      00fh
0006'  0F                                db      00fh
0007'  0F                                db      00fh
0008'  0F                                db      00fh
0009'  0F                                db      00fh
000A'  0F                                db      00fh
000B'  0F                                db      00fh
000C'  0F                                db      00fh
000D'  0F                                db      00fh
000E'  0F                                db      00fh
000F'  0F                                db      00fh
0010'  0F                                db      00fh
0011'  0F                                db      00fh
0012'  0F                                db      00fh
0013'  0F                                db      00fh
0014'  0F                                db      00fh
0015'  0F                                db      00fh
0016'  0F                                db      00fh
0017'  0F                                db      00fh
0018'  0F                                db      00fh
0019'  0F                                db      00fh
001A'  0F                                db      00fh
001B'  0F                                db      00fh
001C'  0F                                db      00fh
001D'  0F                                db      00fh
001E'  0F                                db      00fh
001F'  0F                                db      00fh

```

## SEMICONDUCTOR IDENTIFICATION

## ROM Codes

VIOSEL - video i/o select prom

0020'	0F	db	00fh
0021'	0F	db	00fh
0022'	0F	db	00fh
0023'	0F	db	00fh
0024'	0F	db	00fh
0025'	0F	db	00fh
0026'	0F	db	00fh
0027'	0F	db	00fh
0028'	0F	db	00fh
0029'	0F	db	00fh
002A'	0F	db	00fh
002B'	0F	db	00fh
002C'	0F	db	00fh
002D'	0F	db	00fh
002E'	0F	db	00fh
002F'	0F	db	00fh
0030'	0F	db	00fh
0031'	0F	db	00fh
0032'	0F	db	00fh
0033'	0F	db	00fh
0034'	0F	db	00fh
0035'	0F	db	00fh
0036'	0F	db	00fh
0037'	0F	db	00fh
0038'	0F	db	00fh
0039'	0F	db	00fh
003A'	0F	db	00fh
003B'	0F	db	00fh
003C'	0F	db	00fh
003D'	0F	db	00fh
003E'	0F	db	00fh
003F'	0F	db	00fh
0040'	0F	db	00fh
0041'	0F	db	00fh
0042'	0F	db	00fh
0043'	0F	db	00fh
0044'	0F	db	00fh
0045'	0F	db	00fh
0046'	0F	db	00fh
0047'	0F	db	00fh
0048'	0F	db	00fh
0049'	0F	db	00fh
004A'	0F	db	00fh
004B'	0F	db	00fh
004C'	0F	db	00fh
004D'	0F	db	00fh

## SEMICONDUCTOR IDENTIFICATION

## ROM Codes

VIOSEL - video i/o select prom

004E'	OF	db	00fh
004F'	OF	db	00fh
0050'	OF	db	00fh
0051'	OF	db	00fh
0052'	OF	db	00fh
0053'	OF	db	00fh
0054'	OF	db	00fh
0055'	OF	db	00fh
0056'	OF	db	00fh
0057'	OF	db	00fh
0058'	OF	db	00fh
0059'	OF	db	00fh
005A'	OF	db	00fh
005B'	OF	db	00fh
005C'	OF	db	00fh
005D'	OF	db	00fh
005E'	OF	db	00fh
005F'	OF	db	00fh
0060'	OF	db	00fh
0061'	OF	db	00fh
0062'	OF	db	00fh
0063'	OF	db	00fh
0064'	OF	db	00fh
0065'	OF	db	00fh
0066'	OF	db	00fh
0067'	OF	db	00fh
0068'	OF	db	00fh
0069'	OF	db	00fh
006A'	OF	db	00fh
006B'	OF	db	00fh
006C'	OF	db	00fh
006D'	OF	db	00fh
006E'	OF	db	00fh
006F'	OF	db	00fh
0070'	OF	db	00fh
0071'	OF	db	00fh
0072'	OF	db	00fh
0073'	OF	db	00fh
0074'	OF	db	00fh
0075'	OF	db	00fh
0076'	OF	db	00fh
0077'	OF	db	00fh
0078'	OF	db	00fh
0079'	OF	db	00fh
007A'	OF	db	00fh
007B'	OF	db	00fh



## SEMICONDUCTOR IDENTIFICATION

## ROM Codes

VIOSEL - video i/o select prom

007C'	0F	db	00fh
007D'	0F	db	00fh
007E'	0F	db	00fh
007F'	0F	db	00fh
0080'	0F	db	00fh
0081'	0F	db	00fh
0082'	0F	db	00fh
0083'	0F	db	00fh
0084'	0F	db	00fh
0085'	0F	db	00fh
0086'	0F	db	00fh
0087'	0F	db	00fh
0088'	0F	db	00fh
0089'	0F	db	00fh
008A'	0F	db	00fh
008B'	0F	db	00fh
008C'	0F	db	00fh
008D'	0F	db	00fh
008E'	0F	db	00fh
008F'	0F	db	00fh
0090'	0F	db	00fh
0091'	0F	db	00fh
0092'	0F	db	00fh
0093'	0F	db	00fh
0094'	0F	db	00fh
0095'	0F	db	00fh
0096'	0F	db	00fh
0097'	0F	db	00fh
0098'	0F	db	00fh
0099'	0F	db	00fh
009A'	0F	db	00fh
009B'	0F	db	00fh
009C'	0F	db	00fh
009D'	0F	db	00fh
009E'	0F	db	00fh
009F'	0F	db	00fh
00A0'	0F	db	00fh
00A1'	0F	db	00fh
00A2'	0F	db	00fh
00A3'	0F	db	00fh
00A4'	0F	db	00fh
00A5'	0F	db	00fh
00A6'	0F	db	00fh
00A7'	0F	db	00fh
00A8'	0F	db	00fh
00A9'	0F	db	00fh

## SEMICONDUCTOR IDENTIFICATION

## ROM Codes

VIOSEL - video i/o select prom

00AA'	0F	db	00fh
00AB'	0F	db	00fh
00AC'	0F	db	00fh
00AD'	0F	db	00fh
00AE'	0F	db	00fh
00AF'	0F	db	00fh
00B0'	0F	db	00fh
00B1'	0F	db	00fh
00B2'	0F	db	00fh
00B3'	0F	db	00fh
00B4'	0F	db	00fh
00B5'	0F	db	00fh
00B6'	0F	db	00fh
00B7'	0F	db	00fh
00B8'	0F	db	00fh
00B9'	0F	db	00fh
00BA'	0F	db	00fh
00BB'	0F	db	00fh
00BC'	0F	db	00fh
00BD'	0F	db	00fh
00BE'	0F	db	00fh
00BF'	0F	db	00fh
00C0'	0F	db	00fh
00C1'	0F	db	00fh
00C2'	0F	db	00fh
00C3'	0F	db	00fh
00C4'	0F	db	00fh
00C5'	0F	db	00fh
00C6'	0F	db	00fh
00C7'	0F	db	00fh
00C8'	0F	db	00fh
00C9'	0F	db	00fh
00CA'	0F	db	00fh
00CB'	0F	db	00fh
00CC'	0F	db	00fh
00CD'	0F	db	00fh
00CE'	0F	db	00fh
00CF'	0F	db	00fh
00D0'	0F	db	00fh
00D1'	0F	db	00fh
00D2'	0F	db	00fh
00D3'	0F	db	00fh
00D4'	0F	db	00fh
00D5'	0F	db	00fh
00D6'	0F	db	00fh
00D7'	0F	db	00fh

# SEMICONDUCTOR IDENTIFICATION

---

## ROM Codes

VIOSEL - video i/o select prom

00D8'	05	db	sel6821
00D9'	05	db	sel6821
00DA'	05	db	sel6821
00DB'	05	db	sel6821
00DC'	06	db	sel6845
00DD'	06	db	sel6845
00DE'	03	db	lightpen
00DF'	0F	db	00fh
00E0'	0F	db	00fh
00E1'	0F	db	00fh
00E2'	0F	db	00fh
00E3'	0F	db	00fh
00E4'	0F	db	00fh
00E5'	0F	db	00fh
00E6'	0F	db	00fh
00E7'	0F	db	00fh
00E8'	0F	db	00fh
00E9'	0F	db	00fh
00EA'	0F	db	00fh
00EB'	0F	db	00fh
00EC'	0F	db	00fh
00ED'	0F	db	00fh
00EE'	0F	db	00fh
00EF'	0F	db	00fh
00F0'	0F	db	00fh
00F1'	0F	db	00fh
00F2'	0F	db	00fh
00F3'	0F	db	00fh
00F4'	0F	db	00fh
00F5'	0F	db	00fh
00F6'	0F	db	00fh
00F7'	0F	db	00fh
00F8'	0F	db	00fh
00F9'	0F	db	00fh
00FA'	0F	db	00fh
00FB'	0F	db	00fh
00FC'	0F	db	00fh
00FD'	0F	db	00fh
00FE'	0F	db	00fh
00FF'	0F	db	00fh
0100'	0F	db	00fh
0101'	0F	db	00fh

end

# SEMICONDUCTOR IDENTIFICATION

---

## ROM Codes

VIOSEL - video i/o select prom

Macros:

Symbols:

LIGHTP 0003    SEL682 0005    SEL684 0006

No Fatal error(s)

## SEMICONDUCTOR IDENTIFICATION

## ROM Codes

```
title VRMM Video Ram Mapping Module
cseg
.radix 16
org 0
```

```
; ZDS part number.: 444-127
; release date: 5/21/82
; prom: TBP18s22 (256*8)
; checksum: 7f80
```

```
0:
00' 00 db 00
01' 01 db 01
02' 02 db 02
03' 03 db 03
04' 04 db 04
05' A0 db A0
06' A1 db A1
07' A2 db A2
08' 05 db 05
09' 06 db 06
0A' 07 db 07
0B' 08 db 08
0C' 09 db 09
0D' A3 db A3
0E' A4 db A4
0F' A5 db A5
```

```
10:
10' 0A db 0A
11' 0B db 0B
12' 0C db 0C
13' 0D db 0D
14' 0E db 0E
15' A6 db A6
16' A7 db A7
17' A8 db A8
18' 0F db 0F
19' 10 db 10
1A' 11 db 11
1B' 12 db 12
1C' 13 db 13
1D' A9 db A9
1E' AA db AA
1F' AB db AB
```

```
20:
20' 14 db 14
21' 15 db 15
22' 16 db 16
23' 17 db 17
24' 18 db 18
```

## SEMICONDUCTOR IDENTIFICATION

## ROM Codes

25'	AC	db	AC
26'	AD	db	AD
27'	AE	db	AE
28'	19	db	19
29'	1A	db	1A
2A'	1B	db	1B
2B'	1C	db	1C
2C'	1D	db	1D
2D'	AF	db	AF
2E'	B0	db	B0
2F'	B1	db	B1

30:			
30'	1E	db	1E
31'	1F	db	1F
32'	20	db	20
33'	21	db	21
34'	22	db	22
35'	B2	db	B2
36'	B3	db	B3
37'	B4	db	B4
38'	23	db	23
39'	24	db	24
3A'	25	db	25
3B'	26	db	26
3C'	27	db	27
3D'	B5	db	B5
3E'	B6	db	B6
3F'	B7	db	B7

40:			
40'	28	db	28
41'	29	db	29
42'	2A	db	2A
43'	2B	db	2B
44'	2C	db	2C
45'	B8	db	B8
46'	B9	db	B9
47'	BA	db	BA
48'	2D	db	2D
49'	2E	db	2E
4A'	2F	db	2F
4B'	30	db	30
4C'	31	db	31
4D'	BB	db	BB
4E'	BC	db	BC
4F'	BD	db	BD

50:			
50'	32	db	32
51'	33	db	33
52'	34	db	34

## SEMICONDUCTOR IDENTIFICATION

## ROM Codes

53'	35	db	35
54'	36	db	36
55'	BE	db	BE
56'	BF	db	BF
57'	C0	db	C0
58'	37	db	37
59'	38	db	38
5A'	39	db	39
5B'	3A	db	3A
5C'	3B	db	3B
5D'	C1	db	C1
5E'	C2	db	C2
5F'	C3	db	C3

60:			
60'	3C	db	3C
61'	3D	db	3D
62'	3E	db	3E
63'	3F	db	3F
64'	40	db	40
65'	C4	db	C4
66'	C5	db	C5
67'	C6	db	C6
68'	41	db	41
69'	42	db	42
6A'	43	db	43
6B'	44	db	44
6C'	45	db	45
6D'	C7	db	C7
6E'	C8	db	C8
6F'	C9	db	C9

70:			
70'	46	db	46
71'	47	db	47
72'	48	db	48
73'	49	db	49
74'	4A	db	4A
75'	CA	db	CA
76'	CB	db	CB
77'	CC	db	CC
78'	4B	db	4B
79'	4C	db	4C
7A'	4D	db	4D
7B'	4E	db	4E
7C'	4F	db	4F
7D'	CD	db	CD
7E'	CE	db	CE
7F'	CF	db	CF

80:			
80'	50	db	50

## SEMICONDUCTOR IDENTIFICATION

## ROM Codes

81'	51	db	51
82'	52	db	52
83'	53	db	53
84'	54	db	54
85'	D0	db	D0
86'	D1	db	D1
87'	D2	db	D2
88'	55	db	55
89'	56	db	56
8A'	57	db	57
8B'	58	db	58
8C'	59	db	59
8D'	D3	db	D3
8E'	D4	db	D4
8F'	D5	db	D5

_90:			
90'	5A	db	5A
91'	5B	db	5B
92'	5C	db	5C
93'	5D	db	5D
94'	5E	db	5E
95'	D6	db	D6
96'	D7	db	D7
97'	D8	db	D8
98'	5F	db	5F
99'	60	db	60
9A'	61	db	61
9B'	62	db	62
9C'	63	db	63
9D'	D9	db	D9
9E'	DA	db	DA
9F'	DB	db	DB

_A0:			
A0'	64	db	64
A1'	65	db	65
A2'	66	db	66
A3'	67	db	67
A4'	68	db	68
A5'	DC	db	DC
A6'	DD	db	DD
A7'	DE	db	DE
A8'	69	db	69
A9'	6A	db	6A
AA'	6B	db	6B
AB'	6C	db	6C
AC'	6D	db	6D
AD'	DF	db	DF
AE'	E0	db	E0
AF'	E1	db	E1



# SEMICONDUCTOR IDENTIFICATION

---

## ROM Codes

_B0:			
B0'	6E	db	6E
B1'	6F	db	6F
B2'	70	db	70
B3'	71	db	71
B4'	72	db	72
B5'	E2	db	E2
B6'	E3	db	E3
B7'	E4	db	E4
B8'	73	db	73
B9'	74	db	74
BA'	75	db	75
BB'	76	db	76
BC'	77	db	77
BD'	E5	db	E5
BE'	E6	db	E6
BF'	E7	db	E7

_C0:			
C0'	78	db	78
C1'	79	db	79
C2'	7A	db	7A
C3'	7B	db	7B
C4'	7C	db	7C
C5'	E8	db	E8
C6'	E9	db	E9
C7'	EA	db	EA
C8'	7D	db	7D
C9'	7E	db	7E
CA'	7F	db	7F
CB'	80	db	80
CC'	81	db	81
CD'	EB	db	EB
CE'	EC	db	EC
CF'	ED	db	ED

_D0:			
D0'	82	db	82
D1'	83	db	83
D2'	84	db	84
D3'	85	db	85
D4'	86	db	86
D5'	EE	db	EE
D6'	EF	db	EF
D7'	F0	db	F0
D8'	87	db	87
D9'	88	db	88
DA'	89	db	89
DB'	8A	db	8A
DC'	8B	db	8B
DD'	F1	db	F1
DE'	F2	db	F2
DF'	F3	db	F3

## SEMICONDUCTOR IDENTIFICATION

## ROM Codes

E0:			
E0'	8C	db	8C
E1'	8D	db	8D
E2'	8E	db	8E
E3'	8F	db	8F
E4'	90	db	90
E5'	F4	db	F4
E6'	F5	db	F5
E7'	F6	db	F6
E8'	91	db	91
E9'	92	db	92
EA'	93	db	93
EB'	94	db	94
EC'	95	db	95
ED'	F7	db	F7
EE'	F8	db	F8
EF'	F9	db	F9

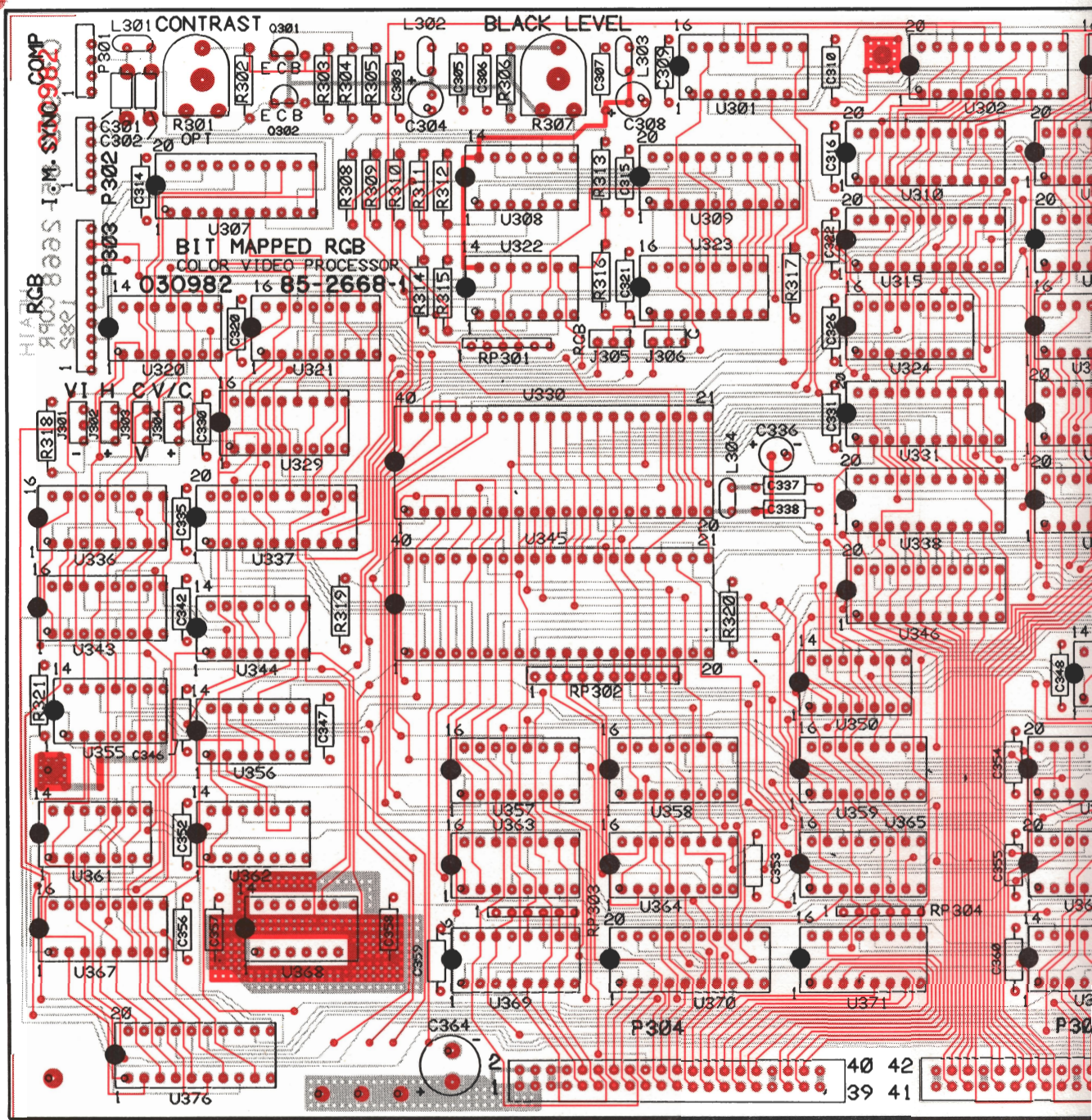
F0:			
F0'	96	db	96
F1'	97	db	97
F2'	98	db	98
F3'	99	db	99
F4'	9A	db	9A
F5'	FA	db	FA
F6'	FB	db	FB
F7'	FC	db	FC
F8'	9B	db	9B
F9'	9C	db	9C
FA'	9D	db	9D
FB'	9E	db	9E
FC'	9F	db	9F
FD'	FD	db	FD
FE'	FE	db	FE
FF'	FF	db	FF

## CIRCUIT BOARD X-RAY VIEW

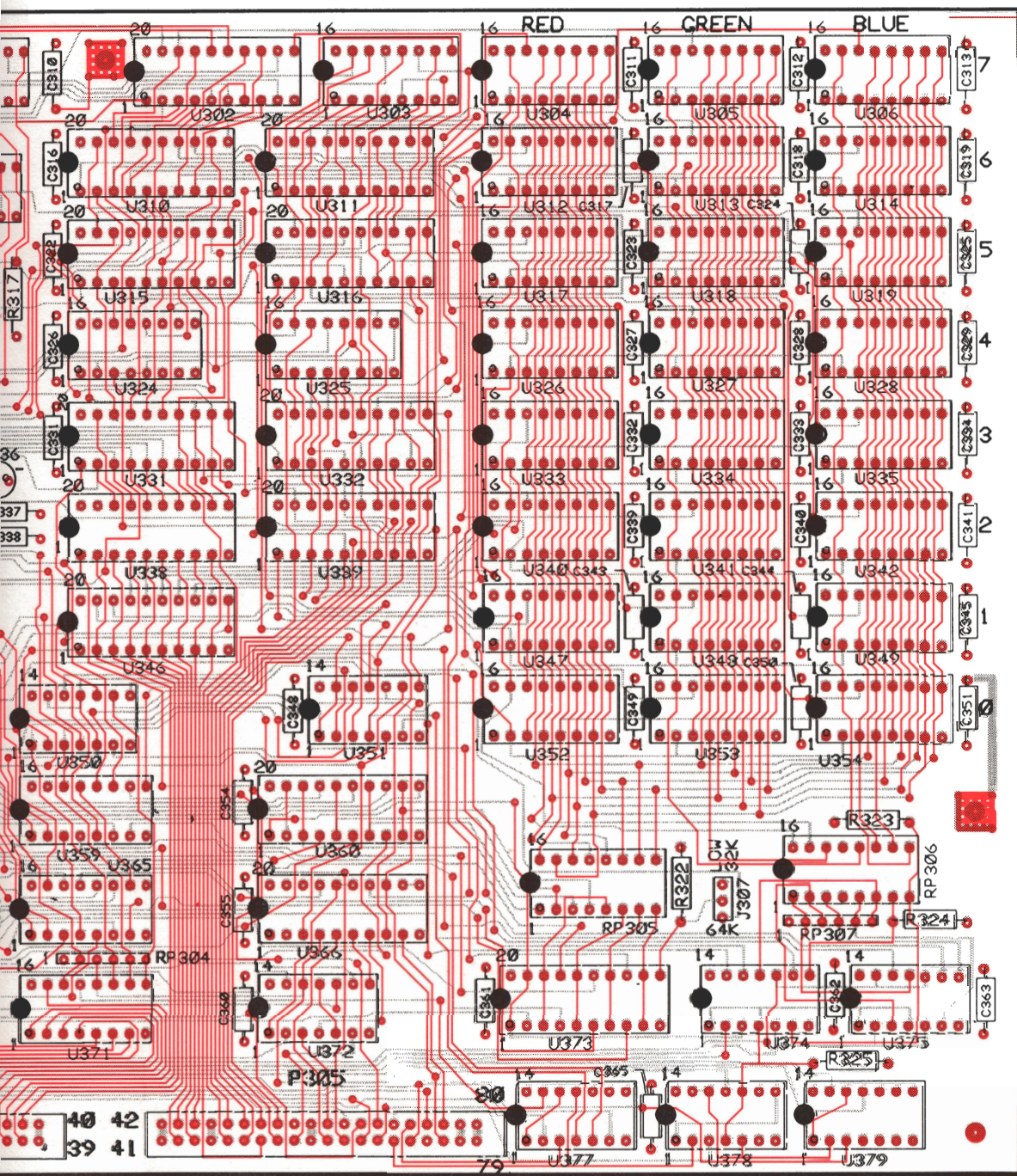
---

**NOTE:** To find the **PART NUMBER** of a component for the purpose of ordering a replacement part:

- A. Find the circuit component number (R303, C304, etc.) on the X-Ray View.
- B. Locate the same number in the "Circuit Component Number" column of the "Replacement Parts List."
- C. Adjacent to the circuit component number, you will find the **PART NUMBER** and **DESCRIPTION** which must be supplied when you order a replacement part.



**VIDEO LOGIC CIRCUIT BOARD**  
 Shown from the component side  
 Component side shown in red, bottom side  
 gray.



**LOGIC CIRCUIT BOARD**  
 from the component side.  
 shown in red, bottom side shown in  
 gray.

## INTERCONNECT PIN DEFINITIONS

---

The following statements briefly define the video logic circuit board connecting pins.

BA0-BA23	Buffered address lines.
BDI0-BDI7	Buffered data input lines.
BDO0-BDO7	Buffered data output lines.
BMWRT	Buffered memory write signal.
$\overline{\text{DBIN}}$	Control signal that requests data on the data input bus.
ECLK	Enable clock signal for the 6845 and the 6821.
GND	Provides common ground for the system.
$\overline{\text{GSEL}}$	Green video RAM select signal.
$\overline{\text{IO}}$	Selects the input or output function.
LTPNSTB	Light pen strobe signal.
MEMR	Memory read status signal.
OUT	Status signal indicating an output data transfer.
$\overline{\text{OUT}}$	Status signal indicating an output data transfer.
$\overline{\text{POC}}$	Power on clear.
RDBFRENBL	Read buffer enable signal.

## INTERCONNECT PIN DEFINITIONS

---

$\overline{\text{RESET}}$  Reset signal that resets the Computer to its power-on status.

STVAL●SYNC Status valid signal ANDed with the sync signal.

VIDRAMRDY Video RAM ready. Causes the CPU to wait if the CPU attempts to access video RAM while the CRT-C is addressing video RAM.

$\overline{\text{WO}}$  Write status signal.

$\overline{\text{WR}}$  Write control signal.

Some other important video signals are:

BDOTA Blue dot (pixel) data signals.

BLUD0-BLUD7 Blue data output bus from video RAM.

BLUE Blue video signal.

$\overline{\text{BSEL}}$  Blue video RAM select signal.

CAS Column address strobe.

CLRSCRN Clear screen signal.

$\overline{\text{CRTRAMSEL}}$  Video RAM select. Indicates that the CPU wants to access VRAM.

DOTCLK Controls the timing of the entire video logic board.

GDOTA Green dot (pixel) data signal.

GRND0-GRND7 Green data output bus from video RAM.

GREEN Green video signals.

## INTERCONNECT PIN DEFINITIONS

---

HI	+ 5 volts through pullup resistor.
HI1	+ 5 volts through pullup resistor.
HI2	+ 5 volts through pullup resistor.
HI3	+ 5 volts through pullup resistor.
HSYNC	Horizontal sync signal.
RAS	Row address stobe signal.
RDOTA	Red dot (pixel) data signals.
RED	Red video signals.
REDD0-REDD7	Red data output bus from video RAM.
$\overline{\text{RSEL}}$	Red video RAM select signal.
VA0-VA7	VRAM address lines.
VERT	Vertical sync signal.
VIDRAMSEL	Video RAM select signal. Indicates CPU has accessed VRAM.
VSYNC	Vertical sync signal.
VSYNC/CSYNC	Vertical sync or composite sync signal. Is selected by jumper.
$\overline{\text{WRTB}}$	Write blue, enables simultaneous write to blue plane.
$\overline{\text{WRTG}}$	Write green, enables simultaneous write to green plane.
$\overline{\text{WRTR}}$	Write red, enables simultaneous write to red plane.





# Video Deflection Board

Circuit Description .....	5.2
Troubleshooting .....	5.4
Recalibration .....	5.5
Replacement Parts List .....	5.8
Circuit Board X-Ray Views .....	5.11
Schematic .....	(Inside Envelope at rear of manual)

## CIRCUIT DESCRIPTION

---

The video deflection board is only used in the all-in-one models of the Z-100 family of computers. It converts TTL signals coming from the video logic board to the voltages necessary to drive the CRT. The board contains the vertical circuits, horizontal circuits, video amplifier, and the high-voltage power supply.

Refer to the Schematic Diagram as you read the following paragraphs.

### Vertical Circuits

The vertical sync signal couples through capacitor C301 to synchronize the vertical oscillator, transistors Q301 and Q302. The oscillator output is from the emitter of Q301, where the signal is shaped by C303 to help produce a linear sweep.

The oscillator signal is coupled to the base of differential amplifier Q303. Its base acts as the inverting input and its emitter as the noninverting input. The output of the amplifier feeds back to its emitter to ensure good linearity, and the RC network between R312 and R317 set the gain and frequency response of the stage.

The output of Q303 drives the vertical driver and amplifier Q304 through Q307. This stage develops the sweep current through the vertical deflection yoke at TX202A. Q308 ensures a fast vertical retrace.

### Horizontal Circuits

The horizontal sync pulse couples through C101 and is applied to Q104. Q104 amplifies the signal and passes it on to the timer, IC101. Here, the signal is shaped and retimed, and applied to horizontal driver Q102. Q102 couples the signal to Q103 through TX101. R127 and C114 shape the signal while R128 dampens any ringing that may occur. The collector current of Q103 couples through the flyback transformer, the width coil, and the linearity coil to drive the horizontal deflection yoke at TX202B.

## CIRCUIT DESCRIPTION

---

### High Voltage Power Supply

The flyback transformer, TX102, uses the signal coming from Q103 to generate the acceleration voltage for the CRT. This voltage is rectified before it leaves the transformer. The secondary of TX102 also develops focus, blanking, and bias voltages for the CRT through C121, CR106, and CR108.

Also, the secondary of T102 develops bias voltages for the horizontal circuits (+ 12 volts) and the video amplifier (+ 70 volts).

### Video Amplifier

The video amplifier is a cascode amplifier consisting of Q401 and Q402. This circuit has high gain, low noise, and low input and output capacitances.

The video signal enters at the base of Q402. A positive voltage at this point is white information. Q401 and Q402 conduct to make the CRT cathode more negative.

Resistor R412 in the emitter circuit of Q402 sets the overall stage gain, while C403, R413, and L401 set the frequency response.

### Power Supply

Power for the video deflection board is a single 12-volt source from the main power supply.

# TROUBLESHOOTING

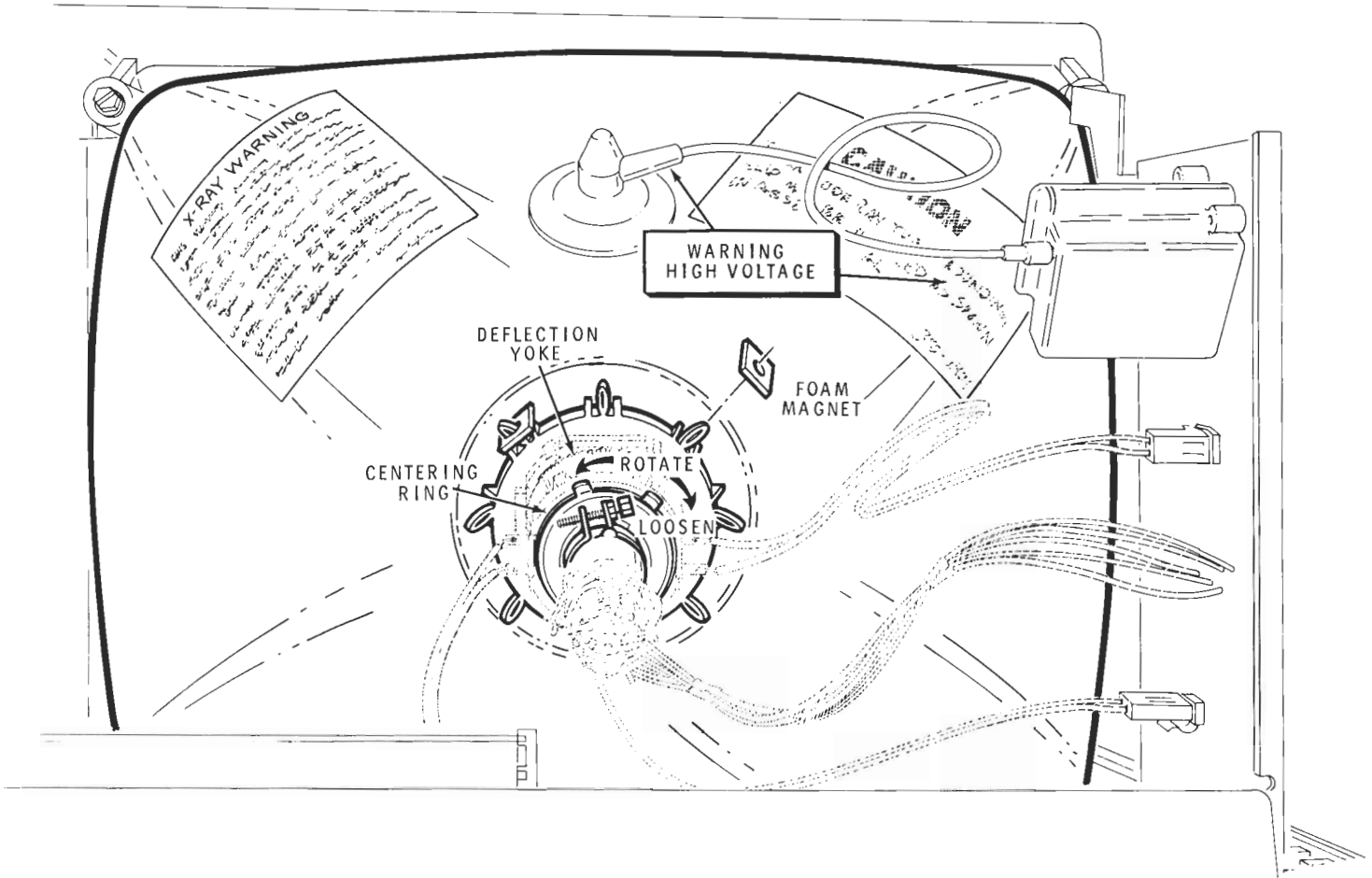
---

Use the following chart for help in identifying the source of problems. The chart lists conditions and possible causes for specific problems. If you cannot resolve the problem, refer to the warranty and service information supplied with your Computer.

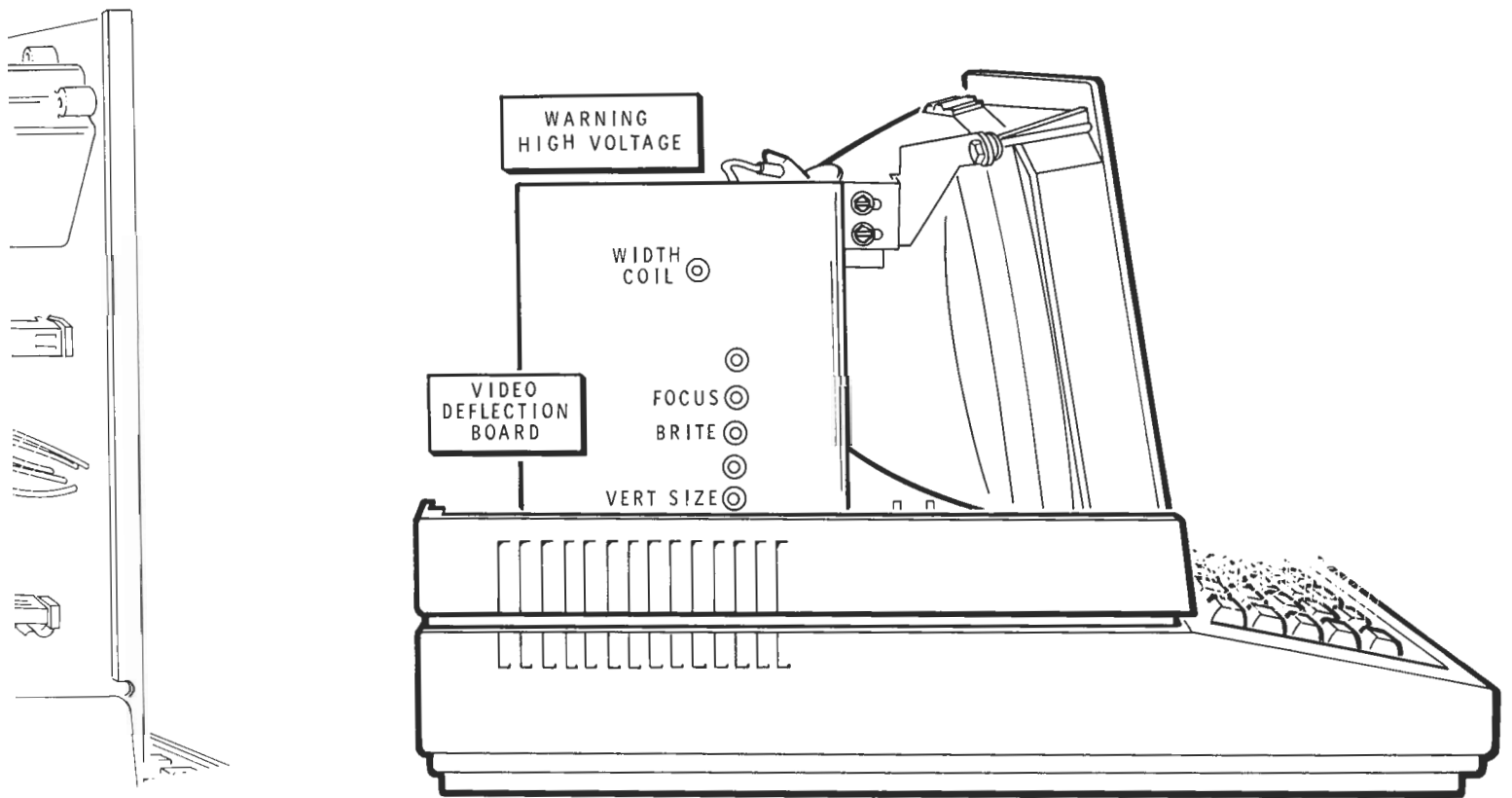
If you have electronics service skill, you may wish to service some problems yourself. In the following chart, if a particular part is mentioned, check that part and other components that are associated with it. Remember to locate and correct the cause when components are damaged, or the problem could reoccur.

Refer to the "Circuit Board X-Ray Views" for the physical location of parts on the circuit boards.

CONDITION	POSSIBLE CAUSE
No high voltage.	<ol style="list-style-type: none"> <li>1. Q102, Q103, or associated circuitry.</li> <li>2. Connector not plugged into vertical deflection board.</li> <li>3. No + 12 volts to deflection board.</li> <li>4. TX102.</li> </ol>
No horizontal sync.	<ol style="list-style-type: none"> <li>1. IC101.</li> <li>2. Q104.</li> <li>3. No timing pulse at base of Q104 (from main board).</li> </ol>
No vertical deflection.	<ol style="list-style-type: none"> <li>1. Q301, Q302, or Q303.</li> <li>2. Q304, Q306, Q307, or associated circuitry.</li> <li>3. Deflection yoke.</li> </ol>
No vertical sync.	<ol style="list-style-type: none"> <li>1. Q302 and associated circuitry.</li> <li>2. No sync signal from main board.</li> </ol>
High voltage present, but no video.	<ol style="list-style-type: none"> <li>1. No video signal from main board.</li> <li>2. Q401, Q402, and associated circuitry.</li> <li>3. Brightness control turned down.</li> </ol>
No focus.	<ol style="list-style-type: none"> <li>1. TX102, R148.</li> <li>2. High voltage is too low.</li> </ol>
Raster (lighted area) is not centered.	<ol style="list-style-type: none"> <li>1. Yoke tabs not adjusted properly.</li> </ol>



**PICTORIAL 5-1**  
**Calibration Control Locat**



**PICTORIAL 5-1**  
**bration Control Locations**

## RECALIBRATION

---

Boot the demo disk supplied with your Computer and utilize the rectangle surrounding the menu for the following procedures.

Refer to Pictorial 5-1 for the following steps.

**NOTE:** In the following adjustments, the controls called for will be on the video deflection circuit board unless stated otherwise. All controls on the circuit board may be accessed from the left side of the computer through holes in the printed circuit board and shield.

**WARNING:** High voltage is present on the back of the CRT and on the video deflection circuit board. As you make adjustments to these areas, use insulated or non metallic tools.

Adjust the BRITE control clockwise until you see the background raster. Then turn the control counterclockwise until the background raster just disappears.

If your Computer has the color memory option, load ZBASIC (or use the Demo Disk) and then enter the following program before proceeding to the next step.

Enter the program exactly as shown:

### COLOR BAR PROGRAM

```
100 CLS
110 COL(1)=1:COL(2)=4:COL(3)=5:COL(4)=2:COL(5)=3:COL(6)=6:COL(7)=7
120 X=1
130 Y=80
140 LINE(0,0)-(640,215),7,B
150 FOR I=0 TO 7
160 LINE (X,1)-(Y,214),COL(I),BF
170 X=X+80
180 Y=Y+80
190 NEXT I
200 END
```



## RECALIBRATION

---

For these adjustments, if you have the color memory option, run the program you have just entered. If you do not have the color memory option, simply follow the instructions in the following steps.

Adjust the rear panel control labeled J14 until the display is at a comfortable brightness level. If you are using the color bar program, and have color RAM installed, you should adjust this control until you can see the eight-step gray scale (black being the first step). Do not make the display too bright as the screen phosphors may be damaged by too much brightness and create 'burns'.

If you have been using the color bar program, return to the demo disk main menu for the following steps.

If necessary, loosen the indicated screw and rotate the deflection yoke until the edges of the display are vertical and horizontal. Then, retighten the screw.

- Adjust the centering rings on the deflection yoke to the position that best centers the rectangle on the screen.
- Adjust the FOCUS control until the characters are as sharp as possible (this may be at one end of the range).
- Adjust the WIDTH coil so the sides of the rectangle are 7/8" to 1-1/8" from the edge at the vertical center (on each side) of the CRT mask. If necessary, recenter the rectangle with the centering rings and check the dimensions again.
- Adjust the VERTICAL SIZE control so the top and bottom of the rectangle are 1/2" (plus or minus 1/8") from the edge of the CRT mask. (If necessary, first temporarily remove the metal rail from the cabinet shell.) Then, if necessary, recenter the rectangle.
- Recheck the dimensions in the preceding two steps and repeat the steps if necessary.

## RECALIBRATION

---

Locate the one area of the four edges of the display that is the least straight. Adjust the foam magnet on the post that protrudes from the yoke at the position which is closest to this location until the display edge is as straight as possible.

Repeat these adjustments as necessary all around the yoke at any of the eight locations which require straightening. The closer the magnets are to the CRT, the greater the effect they will have.

Repeat any of the above adjustments as necessary for an optimum display.

# REPLACEMENT PARTS LIST

---

CIRCUIT	HEATH	Description
Comp. No.	Part No.	

## Resistors

All resistors are 1/4-watt, 5%, unless specified otherwise.

R101	6-102-12	1000 $\Omega$
R103	6-102-12	1000 $\Omega$
R106	6-223-12	22 k $\Omega$
R107	6-102-12	1000 $\Omega$
R109	6-472-12	4700 $\Omega$ , 2%
R112	6-103	10 k $\Omega$ , 1/2-watt, 2%
R116	6-102-12	1000 $\Omega$
RX122	234-282	22 $\Omega$ , failsafe
RX124	1-55-12	10 $\Omega$ , 1/2-watt, failsafe
R127	6-181-12	180 $\Omega$
R128	6-820-12	82 $\Omega$
RX129	234-283	100 $\Omega$ , failsafe
R131	6-681-12	680 $\Omega$
R132	6-153-12	15 k $\Omega$
R137	6-103-12	10 k $\Omega$
R138	6-103-12	10 k $\Omega$ , 10%
R139	234-288	100 k $\Omega$ control
R142	6-222	2200 $\Omega$ , 1/2-watt, 10%
R144	6-274	270 k $\Omega$ , 1/2-watt, 10%
R146	6-103-12	10 k $\Omega$ , 1/2-watt, 10%
R147	6-683-12	68 k $\Omega$ , 10%
R148	234-287	2 M $\Omega$ control
R149	6-274	270 k $\Omega$ , 1/2-watt, 10%
R151	6-473-12	47 k $\Omega$
R301	6-562-12	5600 $\Omega$
R302	6-223-12	22 k $\Omega$
R303	6-204-12	200 k $\Omega$
R304	6-470-12	47 $\Omega$
R306	6-273-12	27 k $\Omega$
R307	6-682-12	6800 $\Omega$
R308	6-273-12	27 k $\Omega$
R309	6-225-12	2.2 M $\Omega$
R311	6-115-12	1.5 M $\Omega$
R312	234-289	250 k $\Omega$ control
R313	6-101-12	100 $\Omega$
R314	6-123-12	12 k $\Omega$
R316	6-273-12	27 k $\Omega$
R317	6-222-12	2200 $\Omega$
R318	6-101-12	100 $\Omega$
R319	6-473	47 k $\Omega$ , 1/2-watt
R321	6-222-12	2200 $\Omega$
R322	6-222-12	2200 $\Omega$
RX323	234-281	3.3 $\Omega$ , failsafe
R324	6-221	220 $\Omega$ , failsafe
R326	6-750-12	75 $\Omega$
R327	6-332-12	3300 $\Omega$
R328	6-391-12	390 $\Omega$

# REPLACEMENT PARTS LIST

---

## Resistors (Cont'd.)

R329	6-681-12	680 $\Omega$
R331	6-279-12	2.7 $\Omega$ , 5%
RX333	234-282	22 $\Omega$ , failsafe
R337	6-101-12	100 $\Omega$
R402	1-50-2	820 $\Omega$ , 2-watt
R403	6-102-12	1000 $\Omega$
R404	6-102-12	1000 $\Omega$
R406	6-470-12	47 $\Omega$ , 10%
R407	6-331	330, 1/2-watt, 10%
R409	6-470-12	47 $\Omega$
R412	6-470-12	47 $\Omega$ , 10%
R413	6-220-12	22 $\Omega$ , 10%
R414	6-153-12	15 k $\Omega$
R416	234-282	22 $\Omega$ , failsafe

## Capacitors

C101	234-285	150 pF
C106	27-161	.01 $\mu$ F
C107	27-105	.0068 $\mu$ F
C109	25-928	33 $\mu$ F
C112	27-161	.01 $\mu$ F
C114	27-128	.022 $\mu$ F
CX116	27-27	.022 $\mu$ F
CX117	234-284	10 $\mu$ F
C118	27-128	.022 $\mu$ F
C119	21-43	.001 $\mu$ F
C121	21-43	.001 $\mu$ F
C122	25-928	33 $\mu$ F
C123	25-942	220 $\mu$ F
C124	25-942	220 $\mu$ F
C126	27-161	.01 $\mu$ F
C127	27-161	.01 $\mu$ F
C128	21-43	.001 $\mu$ F
C129	21-43	.001 $\mu$ F
C301	234-286	1500 pF
C302	234-285	150 pF
C303	27-77	.1 $\mu$ F
C304	25-928	33 $\mu$ F
C307	25-917	10 $\mu$ F
C308	25-900	1 $\mu$ F
C309	25-900	1 $\mu$ F
C311	25-917	10 $\mu$ F
C312	25-884	47 $\mu$ F
C313	25-917	10 $\mu$ F
C314	27-128	.022 $\mu$ F
C316	25-905	470 $\mu$ F
C317	25-942	220 $\mu$ F
C401	25-912	3.3 $\mu$ F
C402	25-917	10 $\mu$ F
C403	234-285	150 pF

# REPLACEMENT PARTS LIST

CIRCUIT Comp. No.	HEATH Part No.	Description
----------------------	-------------------	-------------

## Inductors

L101	234-259	Width coil
L102	234-260	Linearity coil

## Transformers

TX101	234-261	Horizontal drive
TX102	234-262	Horizontal sweep

## Diodes

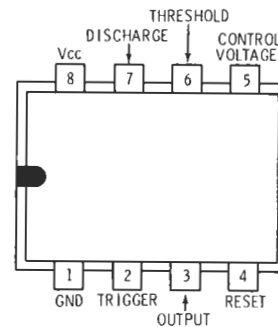
CR102	57-27
CR104	234-264
CR106	234-263
CR107	57-27
CR109	234-265
CR111	234-263
CR112	234-267
CR301	234-266
CR302	57-27
CR303	57-27
CR304	234-267
CR401	234-267

## Transistors

Q102	234-270	Horizontal driver
Q103	234-276	Horizontal output
Q104	234-275	Sync amplifier
Q301	234-275	Vertical oscillator I
Q302	234-274	Vertical oscillator II
Q303	234-274	Differential amplifier
Q304	234-270	Vertical driver
Q306	234-272	Vertical output II
Q307	234-271	Vertical output I
Q308	234-270	Vertical retrace
Q401	234-273	Video output
Q402	234-290	Video driver

## Integrated Circuit

U101	234-269	Timer
------	---------	-------

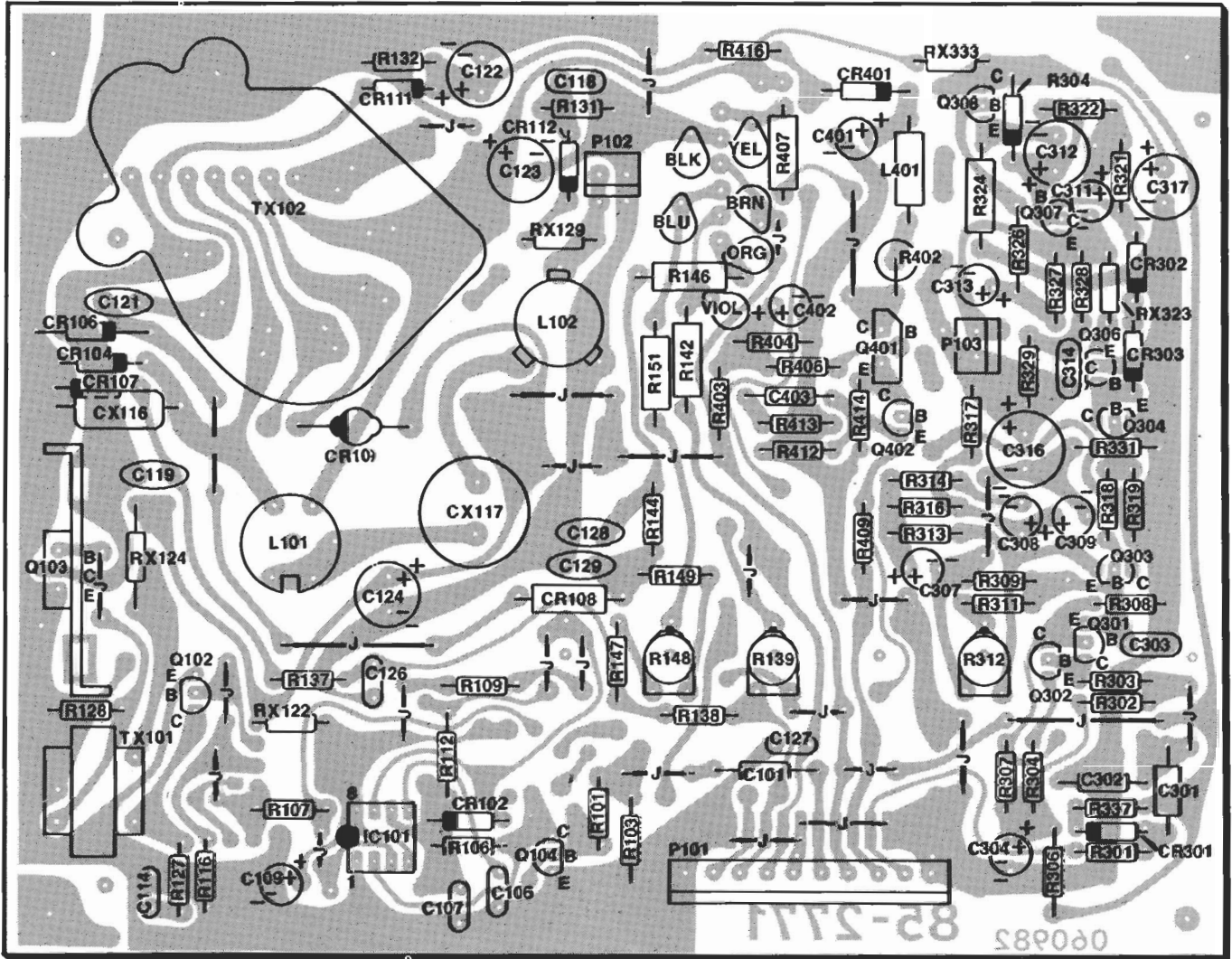


## CIRCUIT BOARD X-RAY VIEWS

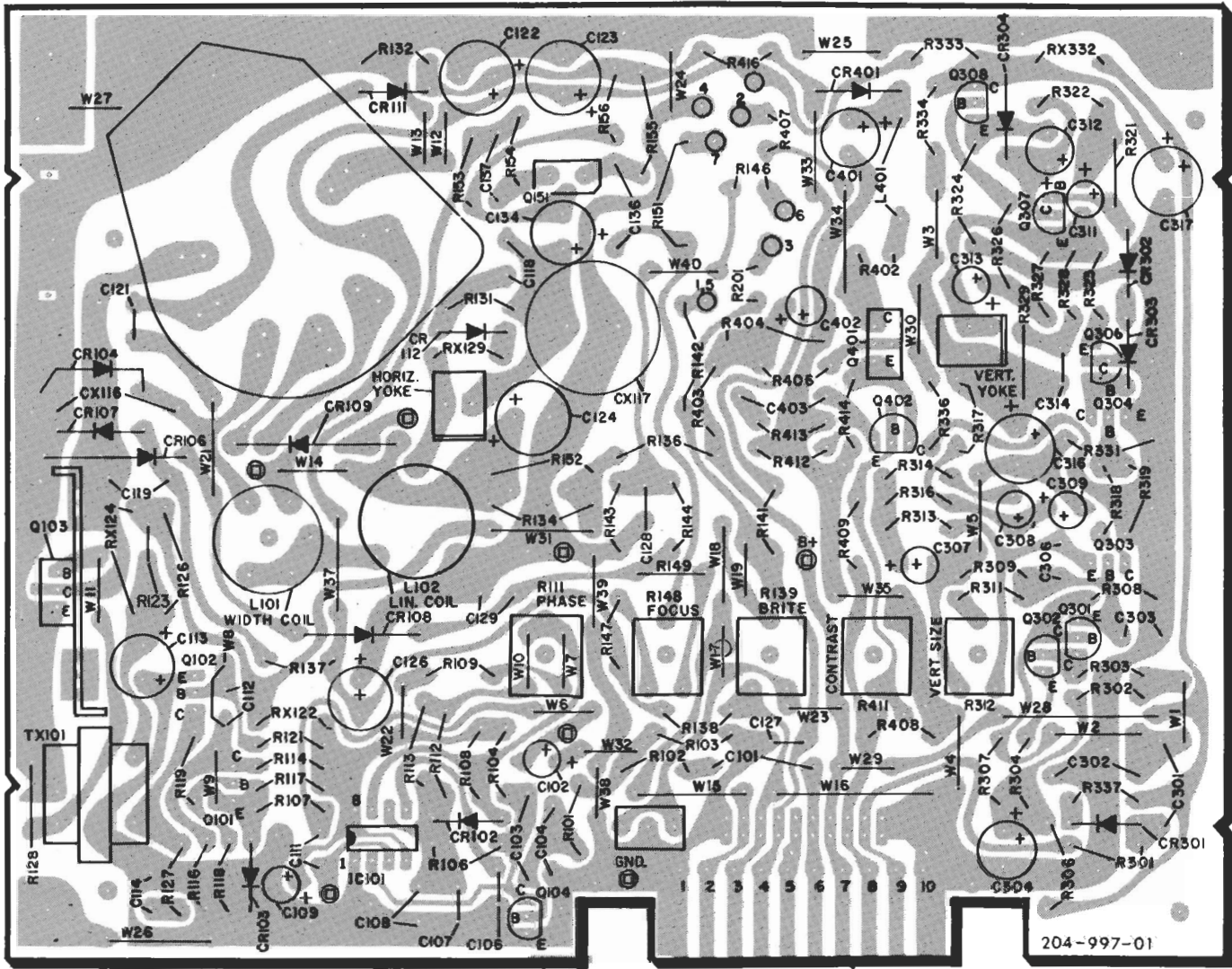
---

**NOTE:** To find the **PART NUMBER** of a component for the purpose of ordering a replacement part:

- A. Find the circuit component number (R303, C304, etc.) on the X-Ray View.
- B. Locate the same number in the "Circuit Component Number" column of the "Replacement Parts List."
- C. Adjacent to the circuit component number, you will find the **PART NUMBER** and **DESCRIPTION** which must be supplied when you order a replacement part.



VIDEO DEFLECTION BOARD (KIT VERSION)



VIDEO DEFLECTION BOARD (WIRED VERSION)



---

# Floppy Disk Controller

Description .....	6.2
User Options .....	6.3
Programming Data .....	6.7
Theory of Operation .....	6.21
Detailed Circuit Description .....	6.23
Troubleshooting .....	6.32
Calibration .....	6.34
Replacement Parts List .....	6.38
Semiconductor Identification .....	6.39
Circuit Board X-Ray View .....	6.49
Interconnect Pin and Signal Definitions .....	6.50
Schematic .....	(Inside Envelope at rear of manual)

## DESCRIPTION

---

The Floppy Disk Controller Card is located in the S-100 card cage in the back of the Z-100 Computer, where it operates as a slave unit on the bus.

The Card has the following features:

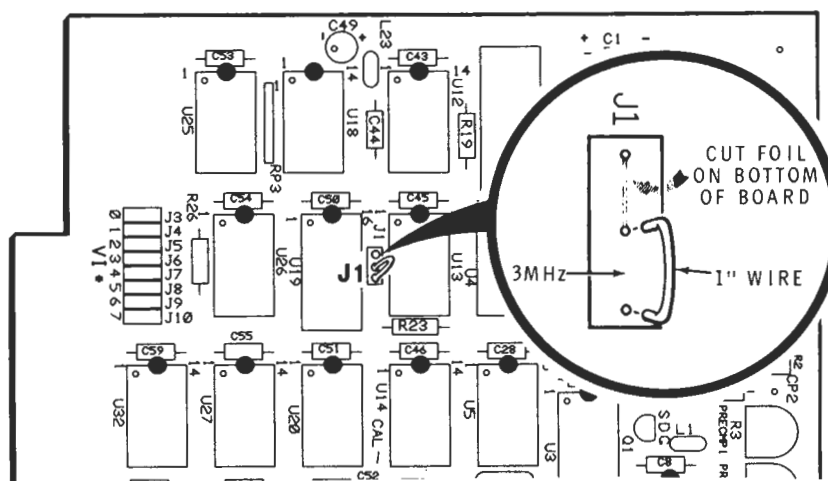
- A user-selectable port address.
- An IEEE 696 S-100 bus compatible interface.
- Up to four 5.25" drives and four 8" drives may be used. (Current software supports only two drives of each type.)
- Single- or double- density, single- or double-sided formats.
- Clock rates up to 5 MHz.
- Stepping rates from 3 to 30 ms.
- Independently adjustable 5" and 8" drive precompensation.
- A phase-locked loop data separator.
- The write signal for the drives is held inactive when the supply voltage drops. (However, due to variations in disk drives, write-protection of disks is not guaranteed when disks are left in the drives during power up or power down.)

# USER OPTIONS

## Card Clock Speed

The Floppy Disk Controller Card is supplied already configured to operate in a Z-100 Family Computer. If the Card will ever be used in a non-standard configuration, then the clock speed jumper may have to be changed as follows:

- If you will be using the Disk Controller with a CPU that operates faster than 3 MHz, no changes are required. The Card is ready for operation.
- If you will be using the Disk Controller with a CPU that operates a 3 MHz or slower, cut the indicated foil on the bottom side of the circuit board at J1 as shown in Pictorial 6-1. Then cut and install a 1" bare wire. Solder the wire ends to the foils.



**Pictorial 6-1**  
Clock Speed Selection

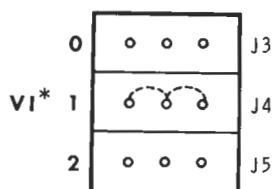
## USER OPTIONS

---

### VI Lines

The Vectored Interrupt lines (VI) are properly configured to operate in a Z-100 Family Computer; no interrupt jumpers are necessary. However, if you use the Controller Card in a non-standard configuration, configure VI lines 0 through 7, as required, by installing the necessary jumper wires. The data request line (DRQ) from the 1797 is connected to holes J3 through J10, while the 1797's interrupt request line (IRQ) is connected to holes 0 through 7. The center row of holes are connected to the S-100 interrupt lines VI0 through VI7, which corresponds to the 0 through 7 numbering of the IRQ holes. Connect the selected option to the proper center hole. See the following example.

Example: A jumper wire soldered from the center hole to J4 selects a data request interrupt on S-100 interrupt line VI1, while a jumper wire soldered from the center hole to 1 selects an interrupt request on S-100 interrupt line VI1. You may connect both interrupt lines to the same center hole if you desire to generate an interrupt on either DRQ or IRQ. See Pictorial 6-2.



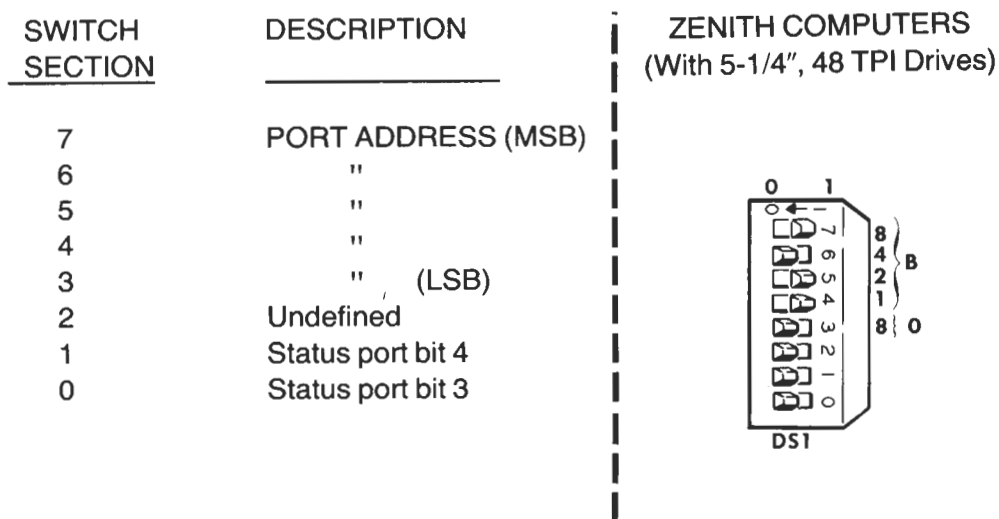
**Pictorial 6-2**  
Selecting Vector Interrupts

## USER OPTIONS

### Port Address Selection

As shown in Pictorial 6-3, the port address is selected by sections 3 through 7 of switch DS1. Switch section 7 selects the most significant bit. Z-100 Family Computers use port address B0 hex as shown.

Switch sections 0 and 1 are used to control bits 3 and 4 (with 0 = least significant bit) of the status port, which can be read at I/O address  $BASE + 5$ . Zenith software currently uses switch section 0 for 48/96 tpi drive selection. The remaining #2 switch position is not used.



**Pictorial 6-3**  
Port Address Selection

## USER OPTIONS

---

### **Other Options**

Other jumpers may be required if you change to different type disk drives and recalibration ever becomes necessary. See the "Calibration" section of this Manual for the use of those jumpers.

## PROGRAMMING DATA

---

This section contains reference tables and data for the programmer who wishes to write software for his Floppy Disk Controller. These tables should be used in conjunction with the 1797 disk controller data sheet (in the rear of this Manual) for complete programming information. Also, several example program segments are given at the end of this section.

### I/O Port Assignments

The following chart lists the I/O Port Addresses of the Floppy Disk Controller Card, while the DIP Switch Definitions chart in Pictorial 6-4 (on Page 6.8) shows how to set the base address of the Card.

#### I/O Ports

<u>I/O ADDRESS (BINARY)</u>	<u>PORT DESIGNATION</u>
BASE + 0	1797 Status register (read-only)
BASE + 0	1797 Command register (write-only)
BASE + 1	1797 Track register
BASE + 2	1797 Sector register
BASE + 3	1797 Data register
BASE + 4	Control latch (write-only)
BASE + 5	Status port (read-only)

NOTE: "BASE" represents the address bits selected on the Floppy Controller's DIP switch.

# PROGRAMMING DATA

---

SWITCH SECTION	DESCRIPTION	ZENITH COMPUTERS (With 5-1/4", 48 TPI Drives)
7	PORT ADDRESS (MSB)	
6	"	
5	"	
4	"	
3	" (LSB)	
2	Undefined	
1	Status port bit 4	
0	Status port bit 3	

**Pictorial 6-4**  
DIP Switch Definitions

## Port Bit Definitions

The definitions of the individual bits written to the 1797 ports listed above are given in the 1797 data sheet in the rear of this Manual.

The control latch bit definitions are given in the following chart. Status port bit definitions are given in the "Status Port Bit Definitions" chart.



# PROGRAMMING DATA

## Control Latch Bit Definitions

BIT NO.	SIGNAL NAME	FUNCTION
0,1	DSA,DSB	00 = Select drive 1 01 = Select drive 2 10 = Select drive 3 11 = Select drive 4
2	8"/5"	0 = Select 5.25" 1 = Select 8"
3	DSEN	0 = Deselect all drives 1 = Select drive specified by bits 0, 1, and 2
4	PRECOMP  5.25" DDEN  8" DDEN  (Note: Precompensation is disabled in single-density.)	0 = Precomp on* 1 = Precomp off  0 = Precomp all tracks 1 = Precomp tracks 44-76
5	5"FASTEP	0 = 1797 operates as specified by bit 2 1 = 1797 operates in 8" mode even for 5" drives
6	WAITEN	0 = Wait state disable 1 = Wait state enable
7	SDEN	0 = Double-density 1 = Single-density

\* Write precompensation is under software control. Heath/Zenith software precompensates tracks 23 and greater.

# PROGRAMMING DATA

---

## Status Port Bit Definitions

BIT NO.	SIGNAL NAME	FUNCTION
0	INTRQ	0 = No interrupt request 1 = Interrupt request from 1797
1	MOTORON (5")	0 = Delay not active running 1 = Delay active
2	DON'T CARE	Not defined
3*	96TPI	Set by section 0 of DIP switch on Floppy Disk Controller Card
4	DON'T CARE	Set by section 1 of DIP switch on Floppy Disk Controller Card.
5	DON'T CARE	Not defined
6	TWOSIDED	0 = 8" Diskette not two-sided 1 = 8" Diskette two-sided
7	DRQ	0 = Not ready for data transfer 1 = Ready for data transfer

\* 0 = 5.25" drives are 48 tpi  
1 = 5.25" drives are 96 tpi

# PROGRAMMING DATA

## Precompensation Options

The following chart lists the signal and jumper requirements to implement the desired write precompensation options for each type of diskette format.

### Signal and Jumper Requirements

TYPE OF DRIVE	NO TRACKS	ALL TRACKS	TRACKS>43
8" Double-Density	N/A	$\overline{\text{PRECOMP}} = 0$ J0 = X	$\overline{\text{PRECOMP}} = 1$ J0 = X
5.25", 48 tpi, Double-Density	$\overline{\text{PRECOMP}} = 1$ J0 = X	$\overline{\text{PRECOMP}} = 0$ J0 = X	N/A
5.25", 96 tpi, Double-Density	$\overline{\text{PRECOMP}} = 1$ J0 = INSTALLED	$\overline{\text{PRECOMP}} = 0$ J0 = X	$\overline{\text{PRECOMP}} = 1$ J0 = OUT

NOTE:  $\overline{\text{PRECOMP}}$  is bit 4 of the control latch, X is a "Don't Care," and precompensation is automatically disabled in single-density operation. J0 is a jumper option on the board that is normally not installed (out).

## Track Formats

The recommended track formats for 5.25" drives are:

Single-Density: Ten 256-byte sectors per track

Double-Density: Eight 512-byte sectors per track

## PROGRAMMING DATA

---

The recommended track formats for 8" drives are:

- Single-Density: Twenty-six 128-byte sectors per track
- Double-Density: Twenty-six 256-byte sectors per track
- Extended Density: Eight 1024-byte sectors per track  
(Z-DOS)

(We recommend that track 0, side 0 of a double-density 8" diskette be recorded in single-density in compliance with the IBM double-density format.)

Zenith software conventions currently use the Card's DIP switch section 0 (status port bit 3) to specify 5.25" drive's **track** density (0 = 48 tpi, 1 = 96 tpi).

### **Interleaving Factors**

The Card can read physically contiguous sectors, and sector interleaving is not required with standard Heath/Zenith systems. Custom applications may require interleaving. It is also possible to implement other formats with 128-, 256-, 512-, or 1024- byte sector sizes in custom applications.

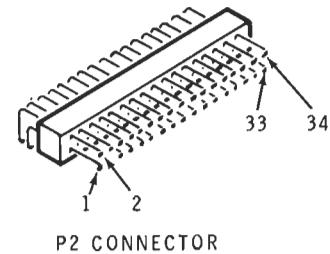
# PROGRAMMING DATA

## Drive Interface Connectors

### 5-1/4" Drive Connector (P2)

NOTE: All signals are active low at the connectors.

PIN No.	DESCRIPTION
1	GND
2 (NC)	Active read filter
3	GND
4 (NC)	TD use control
5	GND
6	Drive select 3
7	GND
8	Index/sector
9	GND
10	Drive select 0
11	GND
12	Drive select 1
13	GND
14	Drive select 2
15	GND
16	Motor on
17	GND
18	Direction select
19	GND
20	Step
21	GND
22	Composite write data
23	GND
24	Write gate
25	GND
26	Track 0
27	GND
28	Write protected
29	GND
30	Composite read data
31	GND
32	Side one select
33	GND
34 (NC)	Disk change



(NC) -- No Connection. These pins are not used by the Controller Card.

# PROGRAMMING DATA

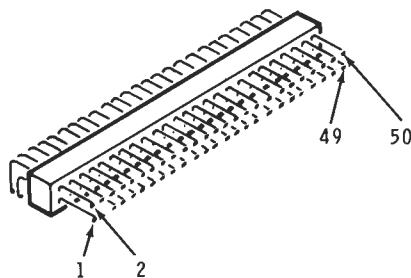
---

## Drive Interface Connectors

### 8" Drive Connector (P1)

NOTE: All signals are active low at the connector.

PIN No.	DESCRIPTION	PIN No.	DESCRIPTION
1	GND	26	Drive select 0
2	Head current switch/ active read filter	27	GND
3	GND	28	Drive select 1
4	(NC) Not assigned	29	GND
5	GND	30	Drive select 2
6	(NC) Not assigned	31	GND
7	GND	32	Drive select 3
8	(NC) Not assigned	33	GND
9	GND	34	Direction select
10	Two-sided	35	GND
11	GND	36	Step
12	(NC) Disk change	37	GND
13	GND	38	Composite write data
14	Side one select	39	GND
15	GND	40	Write gate
16	(NC) In use control	41	GND
17	GND	42	Track 0
18	Head load	43	GND
19	GND	44	Write protected
20	Index	45	GND
21	GND	46	Composite read data
22	Drive ready	47	GND
23	GND	48	(NC) Separated read data
24	(NC) Sector	49	GND
25	GND	50	(NC) Separated read clock



P1 CONNECTOR

(NC) -- No Connection. These pins are not used by the Controller Card.

## PROGRAMMING DATA

## S-100 BUS CONNECTOR

PIN No.	SIGNAL	PIN No.	SIGNAL	PIN No.	SIGNAL
1	+ 8 volts	35	DO1/Data Out 1	69	(NC)RFU
2	+ 16 volts	36	DO0/Data Out 0	70	GND
3	(NC)XRDY	37	(NC)A10	71	(NC)RFU
4	VI0*	38	DO4/Data Out 4	72	RDY
5	VI1*	39	DO5/Data Out 5	73	(NC)INT*
6	VI2*	40	DO6/Data Out 6	74	(NC)HOLD*
7	VI3*	41	DI2/Data In 2	75	RESET*
8	VI4*	42	DI3/Data In 3	76	pSYNC
9	VI5*	43	DI7/Data In 7	77	pWR*
10	VI6*	44	(NC)sM1	78	pDBIN
11	VI7*	45	sOUT	79	A0
12	(NC)NMI*	46	sINP	80	A1
13	(NC)PWRFAIL*	47	(NC)sMEMR	81	A2
14	(NC)DMA3*	48	(NC)sHLTA	82	A6
15	(NC)A18	49	(NC)Clock	83	A7
16	(NC)A16	50	GND	84	(NC)A8
17	(NC)A17	51	+ 8 volts	85	(NC)A13
18	(NC)SDSB*	52	(NC) - 16 volts	86	(NC)A14
19	(NC)CDSB*	53	GND	87	(NC)A11
20	GND	54	(NC)Slave CLR*	88	DO2/Data Out 2
21	(NC)(8088/8085)	55	(NC)DMA0*	89	DO3/Data Out 3
22	(NC)ADSB*	56	(NC)DMA1*	90	DO7/Data Out 7
23	(NC)DODSB*	57	(NC)DMA2*	91	DI4/Data In 4
24	$\Phi$	58	(NC)sXTRQ*	92	DI5/Data In 5
25	pSTVAL*	59	(NC)A19	93	DI6/Data In 6
26	(NC)pHLDA	60	(NC)SIXTN*	94	DI1/Data In 7
27	(NC)RFU	61	(NC)A20	95	DI0/Data In 0
28	(NC)RFU	62	(NC)A21	96	(NC)sINTA
29	A5	63	(NC)A22	97	(NC)sWO*
30	A4	64	(NC)A23	98	(NC)ERROR*
31	A3	65	(NC)NDEF	99	(NC)POC*
32	(NC)A15	66	(NC)NDEF	100	GND
33	(NC)A12	67	(NC)PHANTOM*		
34	(NC)A9	68	(NC)MWRT		

(NC) -- No Connection. These pins are not used by the Controller Card.

# PROGRAMMING DATA

---

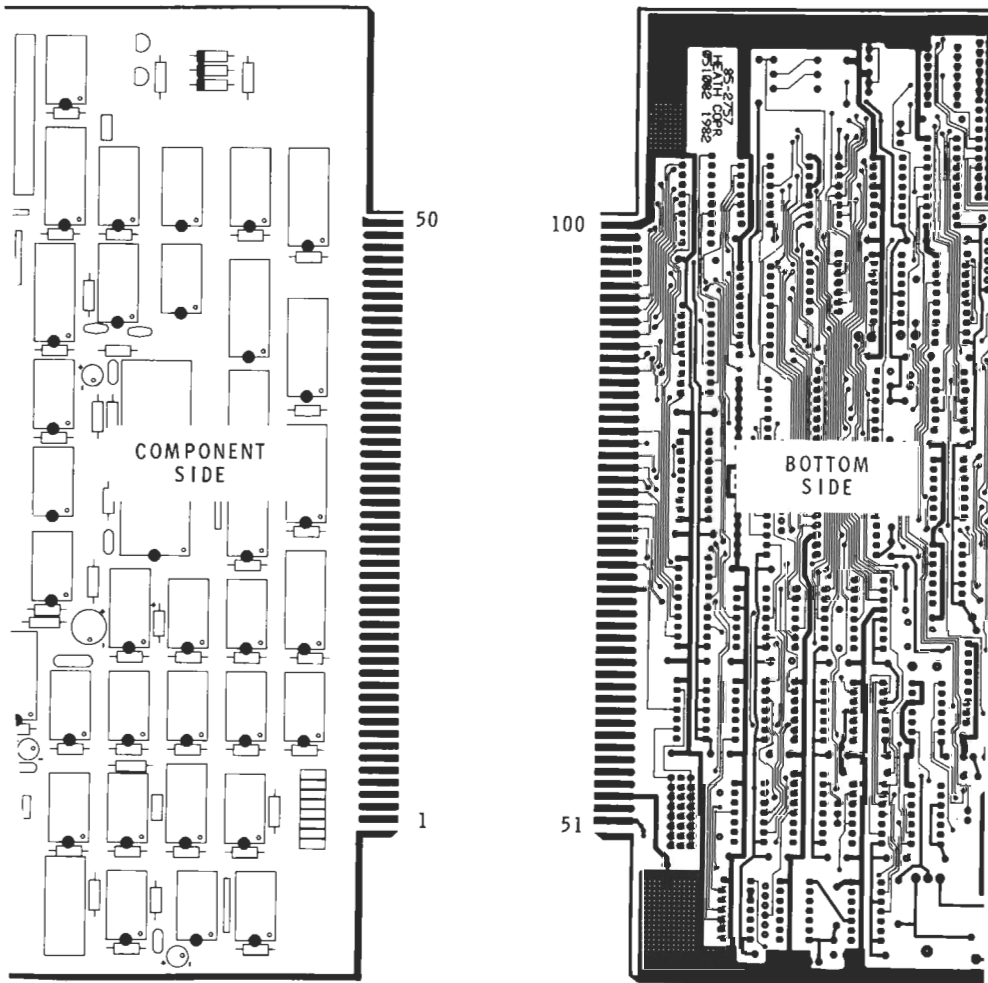
## Sample Programs

```

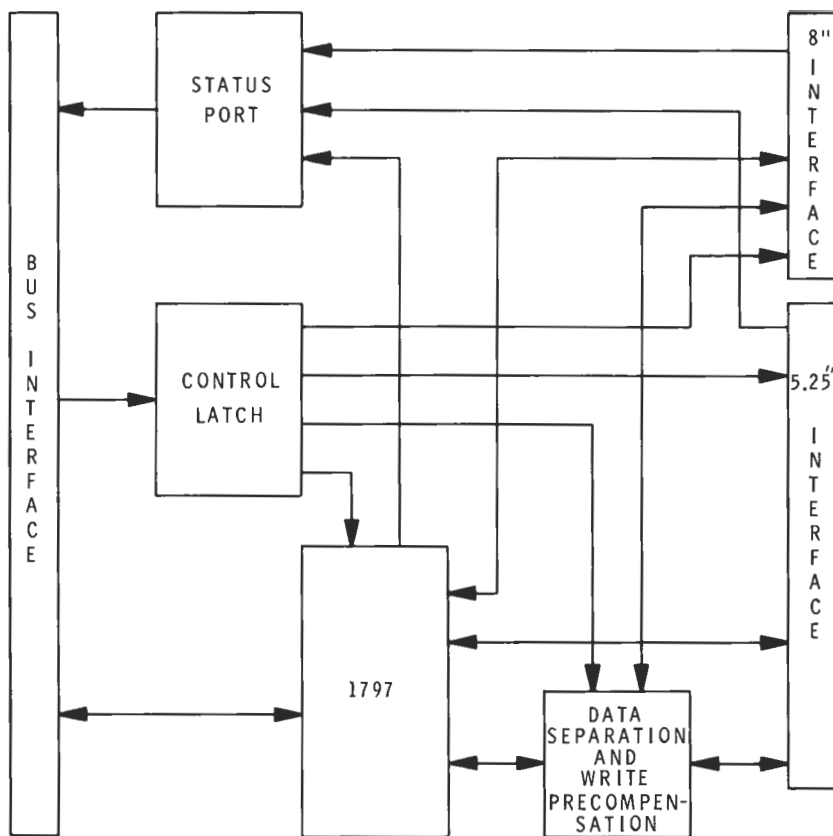
;
; SHOWN HERE ARE EXAMPLES OF THE TYPE OF ASSEMBLY LANGUAGE CODE
; REQUIRED FOR COMMON OPERATIONS WITH THE H/Z-207 DISK CONTROLLER.
; IN ALL CASES, IT IS ASSUMED THAT THE DRIVE, DENSITY AND PRECOMP
; HAVE BEEN SELECTED AND THAT WAIT STATES ARE ENABLED PRIOR TO ANY
; ATTEMPT TO READ/WRITE/SEEK A PARTICULAR DRIVE THROUGH A WRITE OF
; THE APPROPRIATE DATA TO THE H/Z-207 CONTROL PORT (FDCON).
;
; H/Z-207 I/O PORTS
;
00B0 =      BASE      EQU      0B0H      ;BASE CONTROLLER PORT
00B0 =      FDCMD     EQU      BASE      ;1797 COMMAND PORT
00B0 =      FDSTA     EQU      BASE      ;1797 STATUS PORT
00B1 =      FDTRK     EQU      BASE+1    ;1797 TRACK REGISTER
00B2 =      FDSEC     EQU      BASE+2    ;1797 SECTOR REGISTER
00B3 =      FDDAT     EQU      BASE+3    ;1797 DATA REGISTER
00B4 =      FDCON     EQU      BASE+4    ;OUTPUT ONLY CONTROL PORT
00B5 =      FDAS      EQU      BASE+5    ;INPUT ONLY AUX STATUS PORT
;
; BIT DEFINITIONS FOR FDCON
;
0003 =      CONDS     EQU      03H      ;DRIVE SELECT BITS
0004 =      CONDS8    EQU      04H      ;8"/5" DRIVE SELECT
0008 =      CONDS8EN EQU      08H      ;0 = DESELECT ALL DRIVES
0010 =      CONPC     EQU      10H      ;WRITE PRECOMP BIT
;
;      5", 0=PRECOMP ON, 1=OFF, ALL TRKS
;      8", 0=ON, 1=OFF, TKS 0-43 ONLY
;      8" TKS > 43 ARE ALWAYS ON.
0020 =      CON5FS    EQU      20H      ;0=NORMAL, 1 USES 8" 1797 CLOCK FOR 5" DRIVES
0040 =      CONWE     EQU      40H      ;1=WAIT STATES ENABLED FOR DRQ/IRQ
0080 =      CONSD     EQU      80H      ;DENSITY SELECT, 0=DBL (MFM), 1=SGL (FM)
;
; BIT DEFINITIONS FOR FDAS
;
0001 =      ASIRQ     EQU      01H      ;IRQ LINE FROM 1797
0002 =      ASMO      EQU      02H      ;MOTOR ON LINE (FOR 5" DRIVES)
0008 =      ASDS1     EQU      08H      ;DIP SWITCH INPUT, SECTION 0
0010 =      ASDS2     EQU      10H      ;DIP SWITCH INPUT, SECTION 1
0040 =      AS2S      EQU      40H      ;DOUBLE SIDED SIGNAL FROM 8" DRIVES

```





**S100-BUS CONNECTION**



**BLOCK DIAGRAM**

# PROGRAMMING DATA

```

0080 =    ASDRQ   EQU    80H           ;DRQ LINE FROM 1797
          ;
          ; MISC EQUATES
          ;
0100 =    SECSIZ EQU    256           ;ASSUMED SECTOR SIZE FOR THE EXAMPLES
          ;
          ; DATA AREAS
          ;
1000      ORG    1000H           ;ASSUMED DATA AREA FOR EXAMPLES
1000      BUFF   DS    256           ;BUFFER
1100      SECT   DS    1           ;ASSUMED LOCATION OF SECTOR TO READ/WRITE
1101      TRK    DS    1
          ;
0000      PAGE    ORG    0

```

## Read a Sector

```

;
; READ A SECTOR
;
; IT IS ASSUMED THAT THE FLOPPY DISK HEAD HAS BEEN POSITIONED
; OVER THE DESIRED TRACK OF THE FLOPPY DISK PREVIOUSLY AND THAT
; THE TRACK NUMBER IS IN THE 1797 TRACK REGISTER (SEE THE "SEEK DESIRED
; TRACK" EXAMPLE PAGE 6.18). THE SECTOR TO BE READ IS TO BE PLACED
; INTO A BUFFER CALLED BUFF. THE SECTOR SIZE MUST MATCH THE SECTOR
; SIZE AS INDICATED BY THE BYTE WRITTEN INTO THE SECTOR HEADER WHEN THE DISK
; WAS FORMATTED. IN THIS EXAMPLE, THE READ COMMAND IS HARD-CODED AS
; 88H, WHICH IS A SINGLE SECTOR READ WITH NO HEAD LOAD DELAY ON SIDE
; ZERO USING IBM COMPATIBLE SECTOR LENGTH FIELDS. VARIOUS BITS IN THE
; READ COMMAND ALTER THESE PARAMETERS - SEE THE 1797 DATA SHEETS. IN
; TYPICAL APPLICATIONS, IT WILL BE NECESSARY TO "COMPUTE" THE READ
; COMMAND AS PART OF THE READ SECTOR CODE, PARTICULARLY WITH RESPECT
; TO THE HEAD LOAD DELAY AND SIDE SELECT BITS.
;
; CAUTION MUST BE USED WHEN APPLYING THE CODE BELOW TO DOUBLE-DENSITY
; 8" DISKS WITH SLOW CPU'S. THE LOOP USED TO READ A SECTOR REQUIRES
; APPROX 45 CLOCK CYCLES. ONLY ABOUT 12 MICROSECONDS PER BYTE ARE
; AVAILABLE WITH 8" DOUBLE-DENSITY DISKS. THIS CODE WILL WORK WITH
; FAST PROCESSORS, BUT TROUBLE WILL ARISE IF AN ATTEMPT IS MADE TO USE
; THIS WITH A SLOW CPU (FOR EXAMPLE, TO GET 40 CLOCK CYCLES FROM A 2MHZ
; 8080 TAKES 20 MICROSECONDS, WHICH IS MORE THAN THE 12 MICROSECONDS
; AVAILABLE). FOR SECTOR SIZES OF 256 BYTES OR LESS, THE TEST FOR END
; OF SECTOR MAY BE SHORTENED. ALTERNATELY, THE TEST MAY BE OMITTED AND
; AN IRQ INTERRUPT USED TO SIGNAL END OF SECTOR. ON Z-80 AND 8088 PROCESSORS
; THE PROCESS MAY ALSO BE SHORTENED BY USING BLOCK I/O AND LOOP INSTRUCTIONS.
;

```

## PROGRAMMING DATA

```

0000 3A0011  READ:  LDA    SECT          ;GET SECTOR TO READ
0003 D3B2      OUT    FDSEC        ;WRITE SECTOR TO 1797 SECTOR REGISTER
0005 210010    LXI    H,BUFF        ;POINT TO DATA BUFFER TO BE FILLED
0008 110001    LXI    D,SECSIZ        ;SECTOR SIZE IN D,E
000B 3E88      MVI    A,88H          ;READ SECTOR COMMAND (SEE ABOVE)
000D D3B0      OUT    FDCMD        ;LOAD IT INTO 1797 COMMAND REGISTER
000F DBB3      RLOOP:  IN     FDDAT        ;READ DATA (A WAIT IS GENERATED UNTIL DRQ)
0011 77        MOV    M,A            ;PLACE IN MEMORY
0012 23        INX    H            ;INCREMENT POINTER
0013 1B        DCX    D            ;DECREMENT BYTE COUNT
0014 7B        MOV    A,E          ;TEST FOR COUNT=0
0015 B2        ORA    D
0016 C20F00    JNZ    RLOOP          ;CONTINUE TO END OF SECTOR
0019 DBB5      RLOOP1: IN    FDAS        ;READ AUX STATUS
001B E601      ANI    1            ;WAIT FOR IRQ
001D CA1900    JZ     RLOOP1         ;NO IRQ YET
0020 DBB0      RLOOP2: IN    FDSTA        ;READ STATUS
0022 47        MOV    B,A          ;SAVE STATUS
0023 E601      ANI    1            ;WAIT FOR NOT BUSY
0025 C22000    JNZ    RLOOP2         ;1797 STILL BUSY
0028 78        MOV    A,B          ;RESTORE STATUS TO A
;
; AT THE CONCLUSION OF THE OPERATION, THE ACCUMULATOR CONTAINS THE 1797
; STATUS BYTE WHICH MAY BE MASKED AND TESTED TO DETERMINE WHETHER OR NOT
; ANY ERRORS WERE ENCOUNTERED (SEE THE 1797 DATA SHEET). IN THE EVENT OF
; AN ERROR DURING THE SECTOR READ, IRQ WILL BE CONTINUOUSLY SET (IT IS CLEARED
; BY READING FDSTA), WHICH WILL PREVENT THE GENERATION OF WAIT STATES. IN
; THIS CASE THE BUFFER WILL CONTAIN GARBAGE WITH THE STATUS BYTE INDICATING
; THE REASON FOR THE READ FAILURE.
;
;
;
PAGE

```

## PROGRAMMING DATA

## Write a Sector

```

;
; WRITE A SECTOR
;
; THE SECTOR WRITE OPERATION IS SIMILAR TO THE READ OPERATION DESCRIBED
; PREVIOUSLY.  IN THE EXAMPLE BELOW, THE WRITE COMMAND IS HARD CODED AS
; 0A8H.  AS BEFORE, VARIOUS BITS IN THIS COMMAND CONTROL THE WRITE OPERATION
; PARAMETERS AS DESCRIBED IN THE 1797 DATA SHEET, AND IN PRACTICE IT WILL
; BE NECESSARY TO DETERMINE SOME OF THE BITS DYNAMICALLY.  THE OTHER
; COMMENTS MADE FOR THE READ COMMAND ALSO APPLY.
;
; NOTE THAT FORMATTING A TRACK IS ACCOMPLISHED BY USING THE WRITE TRACK
; COMMAND INSTEAD OF THE WRITE SECTOR COMMAND AND WRITING FROM AN EXTREMELY
; LARGE BUFFER WHICH CONTAINS HEADERS, SYNC BYTES, ETC AS WELL AS THE DATA
; FILL CHARACTERS FOR EACH SECTOR ON THE TRACK.  THE CODE IS ESSENTIALLY
; THE SAME AS THAT SHOWN BELOW FOR THE WRITE SECTOR COMMAND, EXCEPT THAT
; A TRACK BYTE COUNT IS USED INSTEAD OF A SECTOR SIZE COUNT..
;
0029 3A0011  WRITE:  LDA      SECT      ;GET SECTOR TO WRITE
002C D3B2          OUT      FDSEC    ;WRITE SECTOR TO 1797 SECTOR REGISTER
002E 210010       LXI      H,BUFF   ;POINT TO DATA BUFFER TO WRITE
0031 110001       LXI      D,SECSIZ  ;SECTOR SIZE IN D,E
0034 3EA8         MVI      A,0A8H   ;WRITE SECTOR COMMAND (SEE ABOVE)
0036 D3B0         OUT      FDCMD    ;LOAD IT INTO 1797 COMMAND REGISTER
0038 7E          WLOOP:   MOV      A,M      ;GET DATA
0039 D3B3         OUT      FDDAT    ;WRITE DATA (A WAIT IS GENERATED UNTIL DRQ)
003B 23          INX      H      ;INCREMENT POINTER
003C 1B          DCX      D      ;DECREMENT BYTE COUNT
003D 7B          MOV      A,E      ;TEST FOR COUNT=0
003E B2          ORA      D
003F C23800       JNZ      WLOOP     ;CONTINUE TO END OF SECTOR
0042 DBB5       WLOOP1:  IN       FDAS    ;READ AUX STATUS
0044 E601         ANI      1        ;WAIT FOR IRQ
0046 CA4200       JZ       WLOOP1    ;NO IRQ YET
0049 DBB0       WLOOP2:  IN       FDSTA   ;READ STATUS
004B 47          MOV      B,A      ;SAVE STATUS
004C E601         ANI      1        ;WAIT FOR NOT BUSY
004E C24900       JNZ      WLOOP2    ;1797 STILL BUSY
0051 78          MOV      A,B      ;RESTORE STATUS TO A
0052 DBB0         IN       FDSTA    ;READ 1797 STATUS
;
; AT THE CONCLUSION OF THE OPERATION, THE ACCUMULATOR CONTAINS THE 1797
; STATUS BYTE WHICH MAY BE MASKED AND TESTED TO DETERMINE WHETHER OR NOT
; ANY ERRORS WERE ENCOUNTERED (SEE THE 1797 DATA SHEET).  IN THE EVENT OF
; AN ERROR DURING THE SECTOR WRITE, IRQ WILL BE CONTINUOUSLY SET (IT IS CLEARED
; BY READING FDSTA), WHICH WILL PREVENT THE GENERATION OF WAIT STATES.  IN
; THIS CASE THE DISK FORMAT WILL NOT BE DAMAGED, ALTHOUGH THE EXACT
; CONSEQUENCES WILL DEPEND ON THE TYPE OF ERROR CONDITION WHICH OCCURED.
;
;
;
;
;
PAGE

```

# PROGRAMMING DATA

---

## Seek a Track

```

;
; SEEK A TRACK
;
; THE SEEK OPERATION (MOVING THE HEAD) IS ACCOMPLISHED BY LOADING THE
; DESIRED TRACK INTO THE DATA REGISTER AND ISSUING THE SEEK COMMAND.
; IN EXECUTING THE SEEK COMMAND, THE 1797 IS ASSUMED TO HAVE THE TRACK
; NUMBER OF THE TRACK OVER WHICH THE HEAD IS CURRENTLY POSITIONED IN THE
; TRACK REGISTER. IF THIS IS NOT THE CASE, THE TRACK REGISTER MAY BE LOADED
; BY WRITING THE CURRENT TRACK TO IT AT ANY TIME. THE TRACK MAY BE DETERMINED
; EITHER BY DOING A READ ADDRESS COMMAND OR BY ISSUING A RESTORE COMMAND,
; WHICH BRINGS THE HEAD TO TRACK ZERO AND LOADS ZERO INTO THE TRACK REGISTER.
;
; AS WAS THE CASE FOR THE READ AND WRITE COMMANDS, THE SEEK COMMAND IS HARD
; CODED IN THIS EXAMPLE TO A 11H. VARIOUS BITS IN THE COMMAND CONTROL
; THE PARAMETERS OF THE SEEK (INCLUDING THE TRACK-TO-TRACK SEEK TIMING),
; AND MAY HAVE TO BE ALTERED IN SPECIFIC APPLICATIONS. IT SHOULD BE
; NOTED THAT SOME DRIVES MAY "SCRIBBLE" ON DISKETTES (WITH UNDESIRABLE
; RESULTS) UNLESS THE SOFTWARE INSURES THAT A SEEK IS NOT PERFORMED UNTIL
; SOME SPECIFIED NUMBER OF MILLISECONDS FOLLOWING THE END OF A WRITE OPERATION.
;
0054 3A0111  SEEK:  LDA    TRK          ;GET TRACK TO MOVE HEAD TO
0057 D3B3      OUT    FDDAT       ;WRITE TRACK TO 1797 DATA REGISTER
0059 3E11      MVI    A,11H        ;SEEK COMMAND
005B D3B0      OUT    FDCMD       ;ISSUE COMMAND TO 1797
005D DBB5      SLOOP1: IN    FDAS        ;READ AUX STATUS
005F E601      ANI    1           ;WAIT FOR IRQ
0061 CA5D00    JZ     SLOOP1      ;NO IRQ YET
0064 DBB0      SLOOP2: IN    FDSTA      ;READ STATUS
0066 47        MOV    B,A         ;SAVE STATUS
0067 E601      ANI    1           ;WAIT FOR NOT BUSY
0069 C26400    JNZ    SLOOP2      ;1797 STILL BUSY
006C 78        MOV    A,B         ;RESTORE STATUS TO A
;
; AT THE CONCLUSION OF THE OPERATION, THE ACCUMULATOR CONTAINS THE 1797
; STATUS BYTE WHICH MAY BE MASKED AND TESTED TO DETERMINE WHETHER OR NOT
; ANY ERRORS WERE ENCOUNTERED (SEE THE 1797 DATA SHEET).
;
006D          END    0

```

## THEORY OF OPERATION

---

Refer to the Block Diagram (Fold-out from Page 6.16), as you read the following description.

The Block Diagram of the Floppy Disk Controller Card consists of seven parts: the bus interface, the status port, the control latch, the 1797 floppy disk controller, the data separation and write precompensation circuitry, and the two drive interfaces.

### **Bus Interface**

The bus interface meets the proposed IEEE 696 standard for an S-100 bus. The bus interface is made up of a connector, two octal bus buffers, an octal tri-state latch, an address comparator, and some miscellaneous enabling circuitry.

### **Status Port**

The status port is a read-only device that tells the CPU the status of the disk drives and the controller. Definitions of the status port bits are listed in the detailed circuit description.

## THEORY OF OPERATION

---

### **Control Latch**

The control latch accepts commands to the disk drives such as DRIVE SELECT, 5" FASTEP, and others that have to do with the selection and mode of the drives. Definitions of the control bits are listed in the detailed circuit description.

### **1797 Controller**

The 1797 controls the placement of information on the diskette. That is, the movement of the drive head, the formation of written data, and the separation of the read data is controlled by the 1797.

### **Data Separation and Precomp**

The data separation and write precompensation circuitry separate data from the clock signal during read operations and precompensate data during double-density write operations.

### **Drive Interfaces**

The 8" and 5.25" drive interfaces include buffers and filter circuitry.



## DETAILED CIRCUIT DESCRIPTION

---

Before you read the rest of this section, you should review the data sheets for the 1797, 1691, and 2143 integrated circuits in Appendix D. Then refer to the schematic drawing while you read the following information.

### S-100 Bus Interface

The S-100 bus interface is compatible with any IEEE 696 S-100 bus. (The bus signal lines are defined in the rear of this Section.)

#### Data In

Data in to the bus (output from the Card) travels through signal lines 91-95 and signal lines 41-43 on the bus interface connector. These pins are used in read operations from the status latch or from the 1797 controller. The data is buffered from the Card's internal data bus to the S-100 bus by means of U36, a 74LS244 buffer.

#### Data Out

Data out from the bus (into the Card) travels through pins 35, 36, 38, 39, 40, 88, 89, and 90 on the bus interface connector. This data is latched by tri-state latch U35. The latch is used because data on an S-100 bus is not held long enough for the 1797 to receive properly. The tri-state latch holds the data on the Card's internal data bus so that the 1797 can read it. Valid data is latched in U35 on every write cycle. The latch is enabled through pin 1 when the ALE (Address Latch Enable) signal latches an asserted sOUT (Status Out) signal via U20.

## DETAILED CIRCUIT DESCRIPTION

---

### **Address Lines**

The address lines from the bus enter the Card through pins 29-31 and 79-83 of the bus interface. They are buffered by the 74LS244 IC, U34.

### **Control Lines**

The control lines from the S-100 bus enter the board through pins 24, 25, 45, 46, and 75-78 of the bus interface. These lines are buffered by U33.

### **Vector Interrupt Lines**

The vector interrupt lines from the bus leave the Card through pins 4 through 11 of the bus interface. They may be driven by U32.

### **Ready Line**

The ready line, RDY, exits through 72 of the bus interface. The line is driven by U32.

### **Power Up**

On power up, the CPU sends a RESET signal to the Floppy Disk Controller Card. This places the 1797 controller, the control latch, the write precompensation control, and the U26 flip-flops in a known state before operation of the Card is attempted.

## DETAILED CIRCUIT DESCRIPTION

---

The reset state for the 1797 is a 03H in the command register, a 01H in the sector register, a 0 in the Not Ready bit (bit 7) of the status register, and a restore command execution. The reset state of the control latch makes all outputs of the latch equal to 0. For the phase lock loop control, the reset state makes the phase four (phi 4) input equal to 0.

Next, the U26 Q outputs are set to 1, which sends an RDY (ready) signal to the CPU and which provides part of the qualification needed for Read and Write enabling through AND gate D of U27.

Also on power up, the WG (write gate) output from the 1797 to the 5.25" and 8" drives is kept high by Q2 and Q3 until the supply voltage is at or above 4 volts at R25. When the supply reaches 4 volts, Q2 and Q3 are biased near their operating region and will conduct when WG is made active at the 1797. This circuitry is designed to prevent accidental writing on diskettes if they are left in the drives when the power is turned on or off. However, diskettes should still not be left in the drives when the power is switched on or off because there is no guarantee that the drives will not accidentally write onto a diskette, without regard to the state of the write gate line.

### **Read and Write Functions**

Reading and writing with the Floppy Disk Controller Card involves transferring three types of information: data which can be read or written, status signals, and control signals. Status signals can only be read and control signals can only be written.

## DETAILED CIRCUIT DESCRIPTION

---

### Read Status Latch (U31)

Assume that a status signal needs to be read. There are two sources of status information for the S-100 bus, the status port and the status register in the 1797. To read the status port, the following happens. The Card is selected by the CPU, which does this by placing the address of the Card on address lines A0-A7. Address lines A3-A7 are checked by the address comparator U29 for the proper address. (The proper address is defined by the user by setting DIP switch DS1.) If the address is proper, U29's EOUT signal is activated on U29's pin 19.

The EOUT signal is gated in U28, NOR gate D, with signal I/O. If signal I/O is low, indicating that the sINP (input) signal or sOUT (status output) signal from the CPU is also present, the simultaneous assertion of EOUT and I/O signals are passed to U20B, a flip-flop whose Q and  $\bar{Q}$  outputs are asserted when the Address Latch Enable (ALE) signal clocks its pins 3 and 11.

The Q output of U20B is ANDed in U27, NAND gate C, with pDBIN, the S-100 data input control signal at pin 78 of the bus interface. The output at pin 8 of U27 becomes low, indicating that the Floppy Disk Controller Card is being read by the CPU, and activates the enable 1 line of the status latch, U31.

The status latch still can not be read until the status port select line (STPS) is asserted at pin 15 of U31. The enable line is activated by U17, the I/O address decoder.

The I/O address decoder activates STPS by decoding address line A0, A1, and A2. If A0 and A1 are low and A2 is high, and if BDSEL or card select is active, the U31's Y1 line is made active. U31 then outputs its status word to the Card's internal data bus, where it is buffered by U36 to the S-100 bus.

## DETAILED CIRCUIT DESCRIPTION

---

### Read Status Register of 1797 (U22)

Assume now that the 1797's status register is to be read. The procedure is the same as the above, except that address lines A0, A1, and A2 are low. Because the address bits A0-A2 are different, the I/O address decoder (U17) does not enable the status latch (U31). Instead, the status register of the 1797 is selected and read onto the data bus.

### Write Control Latch (U30)

The control latch is written at the falling edge of CLEN, which is the simultaneous assertion of pWR and the Y0 output of the I/O address decoder. The pWR signal comes directly from the CPU, and the Y0 signal occurs when A0, A1, and A2 are high, low, and high, respectively. The Y0 and pWR signals are ANDed by U21, gate B. When both Y0 and pWR are active, gate B produces an active low clock, whose trailing edge activates U30.

The control latch receives the control byte from the internal data bus. The control byte is cleared in U30 by a RESET signal from the CPU.

When the WAITEN bit in the control latch is active, a wait state is initiated on the next read or write of the data register. This puts the CPU in a wait state (negates the RDY signal on the S-100 Bus) until DRQ is generated by the disk controller. Upon DRQ becoming active, an additional delay is needed to fulfill the access time requirements of the 1797 Controller IC. The access delay and synchronization to the S-100 bus are both accomplished by counting system clocks. An on-board jumper selects whether three system clocks are counted (for systems with clocks up to 3 MHz) or two system clocks are counted (for systems with clocks up to 5 MHz).

## DETAILED CIRCUIT DESCRIPTION

---

At the completion of the access delay, the wait state is cleared, RDY is asserted, and the CPU completes the read or write of the data register in the 1797. A RESET or an INTRQ signal also clears the wait state, so that the CPU does not hang up after an error during a disk access.

### **Write Command Register in 1797 (U22)**

The command register in the 1797 can be written when A0, A1, and A2 are all low. The FDWR signal is made active when both FDEN and pWR are active low. The signal pWR comes directly from the CPU, while FDEN is a composite signal made up of the FDSEL signal and the signal that starts the access of the 1797 controller at the end of the wait state.

### **Data Read/Write Operations**

**WRITE OPERATIONS.** The Card is enabled by the proper address and by pWR. After the proper control words are sent to select the power drive, address lines A0 and A1 are made high and A2 is made low, connecting the data register of the 1797 to the internal data bus. As long as A0 and A1 are high and A2 and FDWR are low, the data from the S-100 bus will go to the 1797 data register and be sifted out serially with clock pulses inserted between bits on pin 31, and WD line. The track and sector registers of the 1797 hold the location where the data is written on the diskette.

**READ OPERATIONS.** A read operation requires the board to be enabled as described earlier. All steps taken to enable the status port are taken except that the I/O address decoder does not enable the status latch because the address provided by the CPU is not correct for a status read from the latch. Instead, the address lines cause the 1797 to dump the bits in its data register onto the Floppy Disk Controller Card's internal data bus, which connects to the U36 buffer and the S-100 bus.

## DETAILED CIRCUIT DESCRIPTION

---

The 1797 fills its data register from the data shift register, which fills serially from the processed RAWDATA data stream. (RAWREAD data processing is discussed in "Data Separation and Precompensation" on Page 6.30.)

### **RDY Delay**

U19 is a quad flip flop that acts as a delay line for the DRQ signal from the 1797 to the RDY line to the CPU at pin 72 of the S-100 bus interface. The input at D1, pin 4 of U19 is output at Q1 after one clock cycle. Q1 is tied to D2 and is output to Q2 after another clock cycle. Q2 is also tied to U25, gate A, and D3. From gate A, the D2 signal presets flip flop U26, part A. Flip flop U26 qualifies the FDSEL signal to enable read/write operations in anticipation of the RDY line being made active.

From D3 of U19, the DRQ signal is output to Q3, which is connected to D4 and to jumper J1, post G. Post G is connected to post F in 3-MHz operations, which do not need additional delay of the DRQ signal. Instead, the output of Q4, which contains the DRQ signal delayed by three to four clock cycles, is connected to jumper J1, post E. For most 6-MHz operation, J1 is connected between post E and post F. For the Z-100 series of Computers, the Computer's internal timing requires that the 6-MHz jumper be used.

### **Data Shaping**

Data pulses to the drive are reshaped by U16, a one-shot multivibrator, to 400 ns. Raw data from the drive is reshaped to 250 ns.

## **DETAILED CIRCUIT DESCRIPTION**

---

### **Data Separation and Precompensation**

Data separation and precompensation are performed primarily by U1, U3, U5, U4, U16, and U22. Almost all of these two functions are internal to these IC's. Therefore, an understanding of the functions requires a careful study of the IC's data sheets.

The only control a user has over the precompensation functions is in the amount of precompensation involved. You can exercise this control by adjusting R3 and R4.

### **Interrupts**

There are two interrupts that the Floppy Disk Controller Card can generate: the interrupt request (INTRQ) and the data request (DRQ). Both of these interrupts originate from the 1797. The INTRQ signal is sent to indicate a command completion or an error. The DRQ signal is sent to indicate that data will be accepted in response to a disk read or write command.

The interrupts can be detected two ways, as either a vectored interrupt on any of the bus interface pins from 4 to 11, or as a bit set in the status port, U31, which can then be polled by the CPU.

The INTRQ signal also pulls the bus out of an error-caused wait state by making the pin 5 Q output of U26, part A, high.

### **Drive Interfaces**

There are two drive interfaces: one for the 8" drives and one for the 5.25" drives.



## DETAILED CIRCUIT DESCRIPTION

---

### 8" Drive Interface

The 8" drive interface, which is designed for use with a standard 50-pin Shugart-compatible (SA801 or SA851) disk drive, connects to the drives cable through P1. All output signals to the drives are buffered through U8 and U10 except WG and HEADLOAD. The WG signal is sent through transistor Q2, as described in "Power Up" on Page 6.24. The HEADLOAD signal is inverted by the U7 NOR gate C before being transmitted to the drives.

All input signals except READY and TWOSIDED are buffered through U9, part A, when part A is enabled by a high on the 8"/5" line. The READY signal is inverted by U6 NAND gate B, while the TWOSIDED signal is inverted by U6 NAND gate D.

### 5.25" Drive Interface

The 5.25" drive interface connects to the drive cable through the P2. All output signals to the drives are buffered through U11 and U10 except WG and MOTOR. The WG signal is sent through Q3, while MOTOR is sent through U7 gate B, a NOR gate that conducts when either the MOTOR or the MOTOR ON DELAY signal is active. (The MOTOR ON DELAY signal keeps the motor running on a drive after the drive access operation is completed, under the assumption that the first access will be followed shortly by another access. This saves the time it would take for the drive motor to come to speed after it has been selected and before it can be accessed.)

All input signals are buffered through U9, part B, when part B is enabled by the 8"/5" line.

## TROUBLESHOOTING

---

In case of improper operation, check the following items:

- Is a diskette installed in the drive?
- Are all of the cables connected properly at each end?
- Are the jumpers on the Disk Controller Card connected properly?
- Is the Disk Controller Card seated properly in the socket?

If the answer to all of the above questions is yes, and the Card still does not work properly, then you should call:

- Your local Zenith Data Systems Dealer;  
or
- The nearest Authorized Zenith Data Systems Service Center (check the list accompanying this product or look in the yellow pages under "Data Processing Equipment");  
or
- The nearest Heathkit Customer Center;  
or
- Zenith Data Systems, Customer Service Assistance, at (312) 671-7550.

## TROUBLESHOOTING

**IMPORTANT:** Be prepared to furnish the following information. It will be helpful in diagnosing and repairing your unit.

- A. The problem you are having.
- B. The name and model of your computer system.
- C. The system configuration.
- D. Any additional information that will help describe your system.

### Troubleshooting Chart

If you want to service your Card yourself instead of sending it to Zenith or Heath for servicing, check the chart below for possible causes to the problems your Card may be having.

PROBLEM	POSSIBLE CAUSE
Drive access light does not turn on when diskette is booted.	<ol style="list-style-type: none"> <li>1. Check for proper connections of floppy cable inside Computer.</li> <li>2. Check for correct placement of Disk Controller in bus connector.</li> <li>3. Be sure DIP switch on Disk Controller is set at the correct address.</li> <li>4. Check positions of P1 and P2 on the Disk Controller.</li> <li>5. Be sure drive 1 is jumpered for drive 1 selection.</li> <li>6. Verify a properly configured and compatible disk drive.</li> </ol>
All diskette access lights turn on and remain on.	<ol style="list-style-type: none"> <li>1. Drive cable is connected with marked edge on the wrong side.</li> <li>2. Drives configured incorrectly.</li> </ol>
Two drives turn on when a boot operation is selected.	<ol style="list-style-type: none"> <li>1. Two drives have their selection jumpers programmed the same.</li> </ol>
Computer will not accept boot command, returns to hand prompt, or starts to boot but does not return to hand prompt without reset.	<ol style="list-style-type: none"> <li>1. Be sure diskette is bootable.</li> <li>2. Be sure diskette is installed in selected drive before boot command is given.</li> <li>3. Be sure DIP switch on Disk Controller is set at the correct address.</li> <li>4. Be sure drive 1 is jumpered for drive 1 selection.</li> <li>5. Be sure DIP switch bits 0 and 1 are selected for the type of drive being used.</li> </ol>

## CALIBRATION

---

If you have an assembled Disk Controller Card, it has been calibrated at the factory to operate properly with Zenith Data Systems (ZDS) and Heath disk drives. Therefore, if you are using ZDS/Heath Equipment, you probably will not need to recalibrate your assembled Controller Card. However, if your Card is accidentally uncalibrated, or if you are not using ZDS/Heath equipment, follow the procedures below.

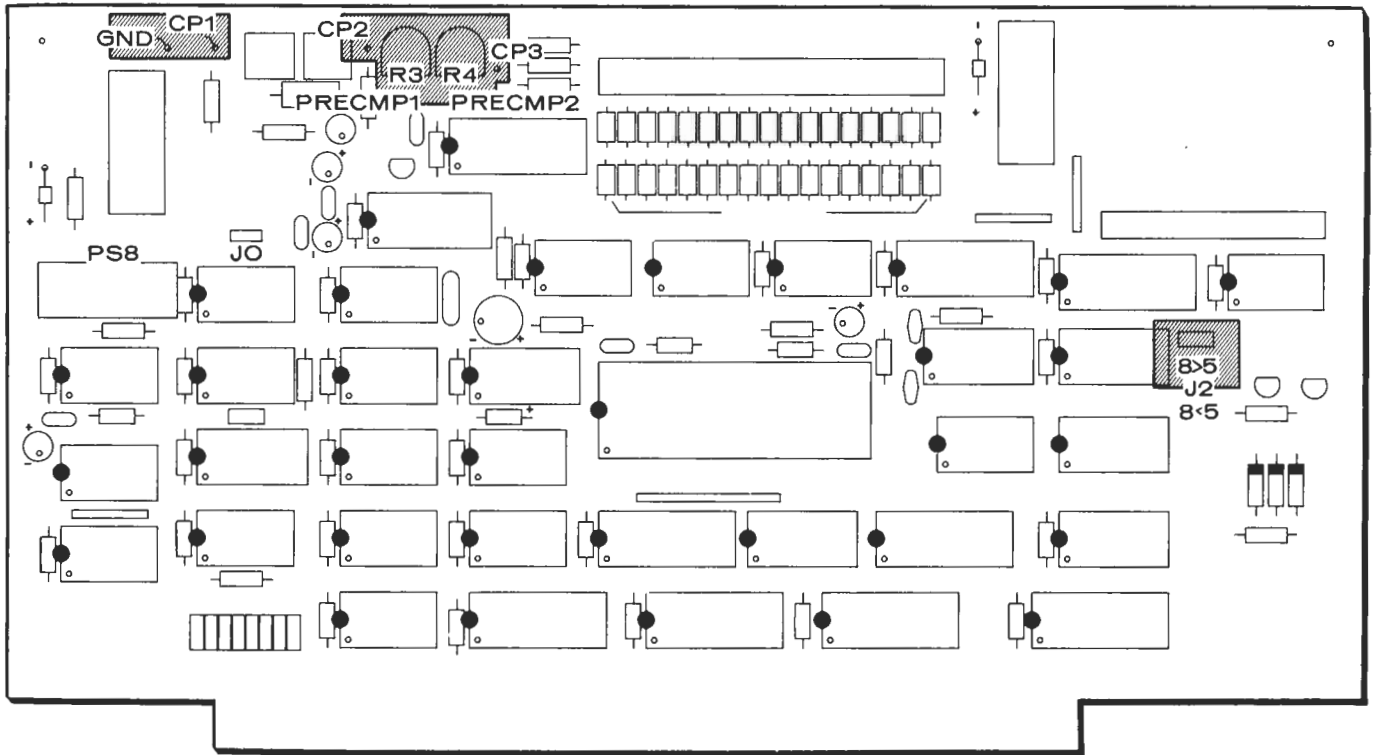
### Equipment Needed

You will need the following equipment to most precisely calibrate your Disk Controller Card:

- A digital voltmeter (DVM) with at least a four-digit readout.
- A 10 MHz bandwidth, calibrated, laboratory-quality oscilloscope with a sweep speed of 50 ns./division and a vertical deflection of 2 V/division, and a low capacitance (X10) probe.
- A frequency counter capable of six-digit accuracy at 4 MHz.
- A blank 8" diskette (or a 5.25" diskette if you are using only 5.25" drives in your system).

### Precompensation Calibration

Usually, two values of precompensation are needed: one for 5.25" drives and one for 8" drives. Accordingly, there are two precompensation adjustment screws on the Disk Controller Card. Potentionmeter R4 is used to set the higher value of precompensation, and potentionmeter R3 is used to set the lower value of precompensation. Pictorial 6-5 (Fold-out from Page 6.34), shows the locations of these two potentionmeters.



**PICTORIAL 6-5**  
**Calibration Locations**

## CALIBRATION

---

Jumper J2 selects whether the 5.25" or the 8" drive will receive the lower value of precompensation. Pictorial 6-5 shows the location of J2 on the Card.

Perform the calibration as follows:

1. Turn the Computer off and remove the Disk Controller Card.
2. Turn R3 fully counterclockwise and R4 fully clockwise.
3. Insert the Disk Controller into the S-100 bus and turn the power on. Allow the Computer to warm up for five minutes.
4. Attach the oscilloscope's probe to CP3 and the probe's ground clip to GND. See Pictorial 6-5.
5. Determine the values of write precompensation that the 5.25" and 8" drives need (the manufacturers of the drives should supply this information with their product). If the value of precompensation is higher for the 5.25" drives, or if you only have 5.25" drives, go to Step 9. If the value of precompensation is higher for the 8" drives, or if you have only 8" drives, go to Step 6. All Heath/Zenith floppy drives require 120 ns of write precompensation.
6. Format a blank 8" diskette in any of the 8" drives by running the FORMAT program provided on your operating system diskette.
7. While FORMAT is running, turn R3 to adjust the pulse width displayed on the oscilloscope to the value of write precompensation needed by your 8" drives. If you do not have 5.25" drives, you have completed the precompensation calibration; proceed to "Data Separator Calibration". If you do have 5.25" drives, continue with the next step.

## CALIBRATION

---

8. Format the 5.25" diskette. While FORMAT is running, turn R4 to adjust the pulse width displayed on the oscilloscope to the value of write precompensation needed by your 5.25" drives. Proceed now to Step 15.
9. If you have both 5.25" and 8" drives, perform the next step. If you have 5.25" drives only, go to Step 11.
10. Cut the foil on the circuit board that connects the middle hole of J2 to the "8<5" hole, the location of which is shown in Pictorial 6-5.
11. Format a blank 5.25" diskette in any of the 5.25" drives by running the FORMAT program provided on your operating system diskette.
12. While FORMAT is running, turn R3 to adjust the pulse width displayed on the oscilloscope to the value of write precompensation needed by your 5.25" drives. If you do not have 8" drives, you have completed the precompensation calibration; proceed to "Data Separator Calibration." If you do have 8" drives, go to the next step.
13. Format the blank 8" diskette.
14. While FORMAT is running, turn R4 to adjust the pulse width displayed on the oscilloscope to the value of write precompensation needed by your 8" drives.
15. Remove the oscilloscope probe.

This completes the precompensation calibration.

## CALIBRATION

---

### Data Separator Calibration

Perform the calibration as follows:

1. Turn the Computer on. Allow at least five minutes for the Disk Controller Card to reach operating temperature.
2. Make sure the disk drives are not selected.
3. Set the DVM's voltage range to 2 V. Attach the common lead to GND and the positive lead to CP2.
4. Adjust R2 (shown in Pictorial 6-5) for a reading of 1.400 V.
5. Remove the voltmeter test leads.
6. Set the six-digit frequency counter to count 4 MHz.
7. Attach the shield lead to GND and the signal lead to CP1.
8. Adjust R1 (shown in Pictorial 6-5) for a reading of 4.000 MHz.
9. Repeat Steps 2 through 8 until there is no further improvement and the 1.4 V and 4 MHz readings occur simultaneously. There will be some (but not much) interaction between these adjustments.
10. Remove the test leads and turn the Computer off.

This completes the calibration procedure.

**NOTE:** Format the blank diskettes used in this procedure again before you use them for recording files.



# REPLACEMENT PARTS LIST

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
----------------------	-------------------	-------------

## Resistors

All resistors are 1/4 W, 5%, unless specified otherwise.

R1	10-1154	10 k $\Omega$ variable, 1/2 W, 10%
RP1	9-106	10 k $\Omega$ resistor pack, 5 W
R2	10-1180	100 k $\Omega$ variable, 1/2 W, 10%
RP2	9-119	10 k $\Omega$ resistor pack, 5 W
R3	10-1137	2000 $\Omega$ variable, 3/4 W, 20%
RP3-RP4	9-120	150 $\Omega$ resistor pack
R4	10-1137	2000 $\Omega$ control
R5	6-470-12	47 $\Omega$
R6	NOT USED	
R7	6-2540-12	47 k $\Omega$ , 1%
R8	NOT USED	
R9	6-105-12	1 M $\Omega$
R10	6-102-12	1000 $\Omega$
R11	6-392-12	3900 $\Omega$
R12	6-185-12	1800 $\Omega$
R13	6-2502-12	47 k $\Omega$ , 1%
R14	NOT USED	
R15	6-7200-12	720 $\Omega$ , 1%
R16-R17	6-2502-12	25 k $\Omega$ , 1%
R18	6-124-12	120 k $\Omega$
R19	6-2370-12	237 $\Omega$ , 1%
R20	6-105-12	1 M $\Omega$
R21	6-392-12	3900 $\Omega$ , 1%
R22	6-1001-12	2200 $\Omega$ , 1%
R23	6-102-12	1000 $\Omega$
R24	6-124-12	120 k $\Omega$
R24A-R24B	6-101-12	100 $\Omega$
R25-R26	6-102-12	1000 $\Omega$

## Capacitors

C1	25-197	1 $\mu$ F tantalum
C2	NOT USED	
C3	29-71	.47 $\mu$ F, 100 V, 1%

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
----------------------	-------------------	-------------

## Capacitors (cont'd)

C4-C6	25-220	10 $\mu$ F tantalum
C7-C8	21-762	.1 $\mu$ F ceramic
C9-C25	21-785	22 pF ceramic
C26	25-197	1 $\mu$ F tantalum
C27-C28	21-762	.1 $\mu$ F ceramic
C29	20-709	36 pF
C30	25-921	47 $\mu$ F tantalum
C31-C34	21-762	.1 $\mu$ F ceramic
C35	25-220	10 $\mu$ F tantalum
C36-C37	21-762	.1 $\mu$ F ceramic
C38-C39	21-746	180 pF
C40-C47	21-762	.1 $\mu$ F ceramic
C48	21-197	1 $\mu$ F tantalum
C49	25-220	10 $\mu$ F tantalum
C50-C63	21-762	.1 $\mu$ F ceramic

## Inductors

L1	235-229	35 $\mu$ H
L2-L18	475-31	1.22 $\mu$ H
L19-L23	235-229	35 $\mu$ H

## Crystal Oscillator

U18	150-132	4 MHz
-----	---------	-------

## Semiconductors

See "Semiconductor Identification"

## SEMICONDUCTOR IDENTIFICATION

---

This section is divided into two parts; "Component Number Index" and Part Number Index." The first section provides a cross-reference between semiconductor component numbers and their respective Part Numbers. The component numbers are listed in numerical order. The second section provides a lead configuration detail (basing diagram) for each semiconductor Part Number. The Part Numbers in the second section are also listed in numerical order.

### Component Number Index

This index shows the Part Number of each semiconductor.

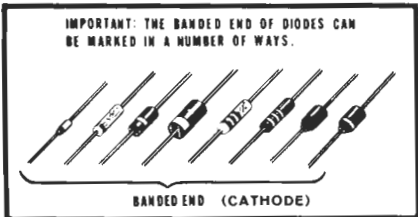
CIRCUIT COMPONENT NUMBER	HEATH PART NUMBER	CIRCUIT COMPONENT NUMBER	HEATH PART NUMBER
D1-D3	56-84	U21	443-875
Q1	417-246	U22	443-997
Q2	417-937	U23	443-798
Q3	417-937	U24	443-877
U1	443-998	U25	443-800
U2	NOT USED	U26	443-900
U3	443-1000	U27	443-728
U4	443-730	U28	443-779
U5	443-999	U29	443-971
U6	443-792	U30	443-805
U7	443-1063	U31	443-1039
U8	443-72	U32	443-72
U9	443-824	U33	443-791
U10	443-753	U34	443-791
U11	443-72	U35	443-863
U12	443-730	U36	443-791
U13	443-811	PS1	442-54
U14	443-730	PS2	442-663
U15	443-1040	PS3	442-708
U16	443-1040		
U17	443-877		
U18	4 MHz oscillator		
U19	443-752		
U20	443-730		

# SEMICONDUCTOR IDENTIFICATION

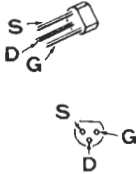

## Part Number Index

This index shows a lead configuration detail (basing diagram) of each semiconductor part number.

### Diodes

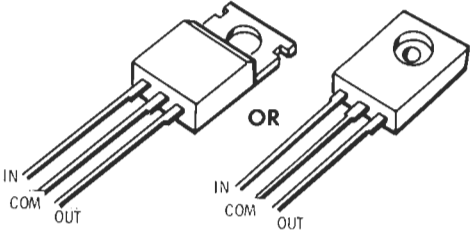
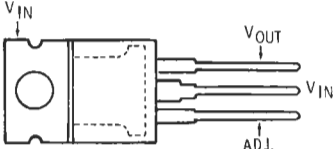
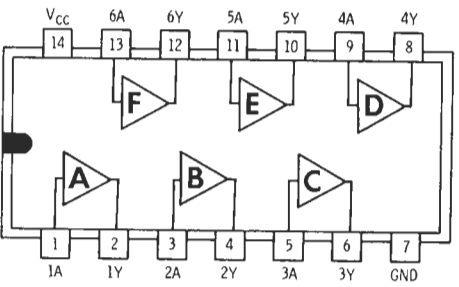
HEATH PART NUMBER	MAY BE REPLACED WITH	DESCRIPTION	LEAD CONFIGURATION (TOP VIEW)
58-84	1N4148	1 mA 75 V SILICON	

### Transistors

HEATH PART NUMBER	MAYBE REPLACED WITH	DESCRIPTION	LEAD CONFIGURATION (TOP VIEW)
417-246	TIS74	FET	
417-937	MPS2369	200 mA 15 V NPN SILICON	

## SEMICONDUCTOR IDENTIFICATION

## Integrated Circuits

HEATH PART NUMBER	MAY BE REPLACED WITH	DESCRIPTION	LEAD CONFIGURATION (TOP VIEW)
442-54	UA 7805	+ 5 V REGULATOR	
442-663	LM 78M12	+ 12 V REGULATOR	
442-708	LM 2904	ADJUSTABLE REGULATOR	
443-72	SN 7417	HEX BUFFERS	

(cont'd)

# SEMICONDUCTOR IDENTIFICATION

## Integrated Circuits (Cont'd)

HEATH PART NUMBER	MAY BE REPLACED WITH	DESCRIPTION	LEAD CONFIGURATION (TOP VIEW)
443-728	74LS00	QUAD NANDS	
443-730	74LS74	DUAL D FLIP-FLOPS	
443-752	74LS175	QUAD D FLIP-FLOPS	
443-753	74S240	OCTAL TRI-STATE BUFFERS	

(cont'd)

# SEMICONDUCTOR IDENTIFICATION

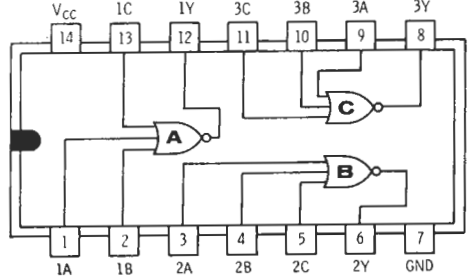
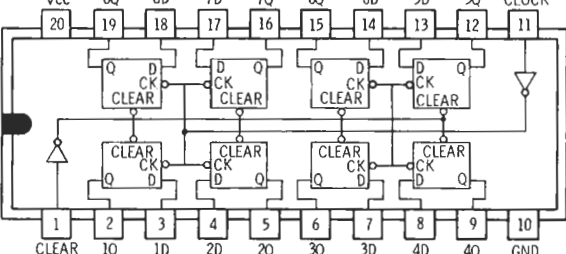
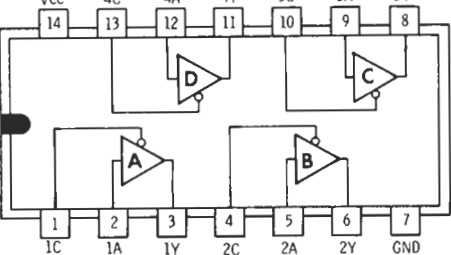
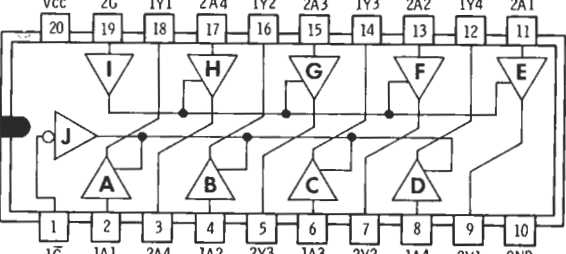
## Integrated Circuits (Cont'd)

HEATH PART NUMBER	MAY BE REPLACED WITH	DESCRIPTION	LEAD CONFIGURATION (TOP VIEW)
443-779	74LS02	QUADNORS	
443-791	74LS244	TRI-STATE BUFFER/DRIVERS	
443-792	74LS132	QUAD NANDS	
443-798	74LS20	DUAL 4-INPUT NANDS	

(cont'd)

# SEMICONDUCTOR IDENTIFICATION

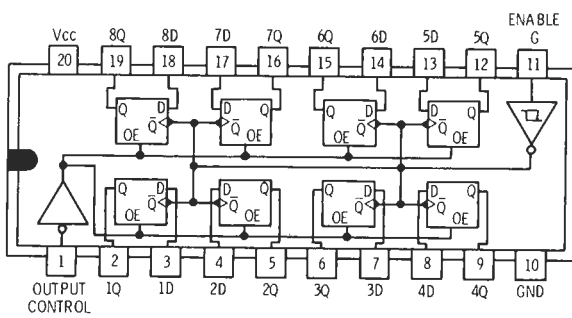
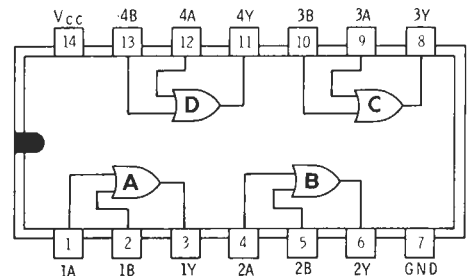
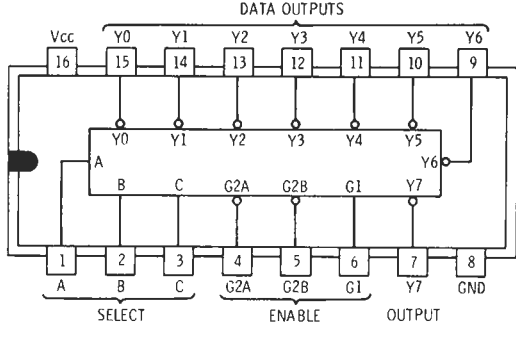
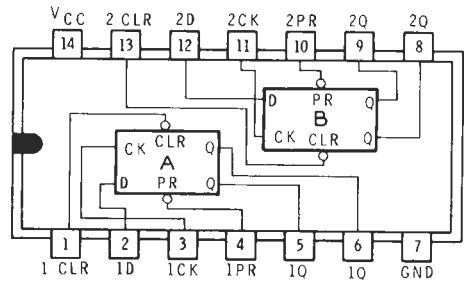
## Integrated Circuits (Cont'd)

HEATH PART NUMBER	MAY BE REPLACED WITH	DESCRIPTION	LEAD CONFIGURATION (TOP VIEW)
443-800	74LS27	TRIPLE 3-INPUT NORS	
443-805	74LS273	OCTAL D FLIP-FLOPS	
443-811	74LS125	QUAD TRI-STATE BUFFER	
443-824	74LS241	TRI-STATE BUFFER/DRIVER	

(cont'd)

# SEMICONDUCTOR IDENTIFICATION

## Integrated Circuits (Cont'd)

HEATH PART NUMBER	MAY BE REPLACED WITH	DESCRIPTION	LEAD CONFIGURATION (TOP VIEW)
443-863	74LS374	OCTAL D TRI-STATE FLIP-FLOP	 <p>The diagram shows an 8-bit octal D-type flip-flop with tri-state outputs. It features 20 pins: Vcc (20), 8Q (19), 8D (18), 7D (17), 7Q (16), 6Q (15), 6D (14), 5D (13), 5Q (12), ENABLE G (11), 1 (1), 1Q (2), 1D (3), 2D (4), 2Q (5), 3Q (6), 3D (7), 4D (8), 4Q (9), and GND (10). Each flip-flop has a D input, a Q output, and an OE (output enable) input. The enable input is active-low and is connected to pin 11.</p>
443-875	74LS32	QUAD 2-INPUT OR	 <p>The diagram shows a quad 2-input OR gate. It has 14 pins: Vcc (14), 4B (13), 4A (12), 4Y (11), 3B (10), 3A (9), 3Y (8), 1A (1), 1B (2), 1Y (3), 2A (4), 2B (5), 2Y (6), and GND (7). Each gate (A, B, C, D) has two 2-inputs and one output.</p>
443-877	74LS138	3-line to 8-line DECODER	 <p>The diagram shows a 3-to-8 line decoder. It has 16 pins: Vcc (16), 15 (15), 14 (14), 13 (13), 12 (12), 11 (11), 10 (10), 9 (9), 8 (8), 7 (7), 6 (6), 5 (5), 4 (4), 3 (3), 2 (2), 1 (1), and GND (8). The select inputs are A (1), B (2), and C (3). The enable inputs are G2A (4), G2B (5), and G1 (6). The output pins are Y0 (15), Y1 (14), Y2 (13), Y3 (12), Y4 (11), Y5 (10), Y6 (9), and Y7 (7).</p>
443-900	74S74	DUAL D FLIP-FLOP	 <p>The diagram shows a dual D-type flip-flop. It has 14 pins: Vcc (14), 2 CLR (13), 2D (12), 2CK (11), 2PR (10), 2Q (9), 2Q (8), 1 CLR (1), 1D (2), 1CK (3), 1PR (4), 1Q (5), 1Q (6), and GND (7). Each flip-flop (A and B) has a D input, a Q output, a CK (clock) input, a CLR (clear) input, and a PR (preset) input.</p>

(cont'd)



# SEMICONDUCTOR IDENTIFICATION

## Integrated Circuits (Cont'd)

HEATH PART NUMBER	MAY BE REPLACED WITH	DESCRIPTION	LEAD CONFIGURATION (TOP VIEW)
443-971	74LS688	8-BIT COMPARATOR	
443-997	1797	FLOPPY DISK CONTROLLER	
443-998	1691	FLOPPY SUPPORT LOGIC	

(cont'd)

## SEMICONDUCTOR IDENTIFICATION

## Integrated Circuits (Cont'd)

HEATH PART NUMBER	MAY BE REPLACED WITH	DESCRIPTION	LEAD CONFIGURATION (TOP VIEW)
443-999	74LS624	VOLTAGE CONTROLLED OSCILLATOR	
443-1000	2143-01	4-PHASE CLOCK GENERATOR	
443-1039	74LS365A	HEX BUFFER	

(cont'd)

# SEMICONDUCTOR IDENTIFICATION

## Integrated Circuits (Cont'd)

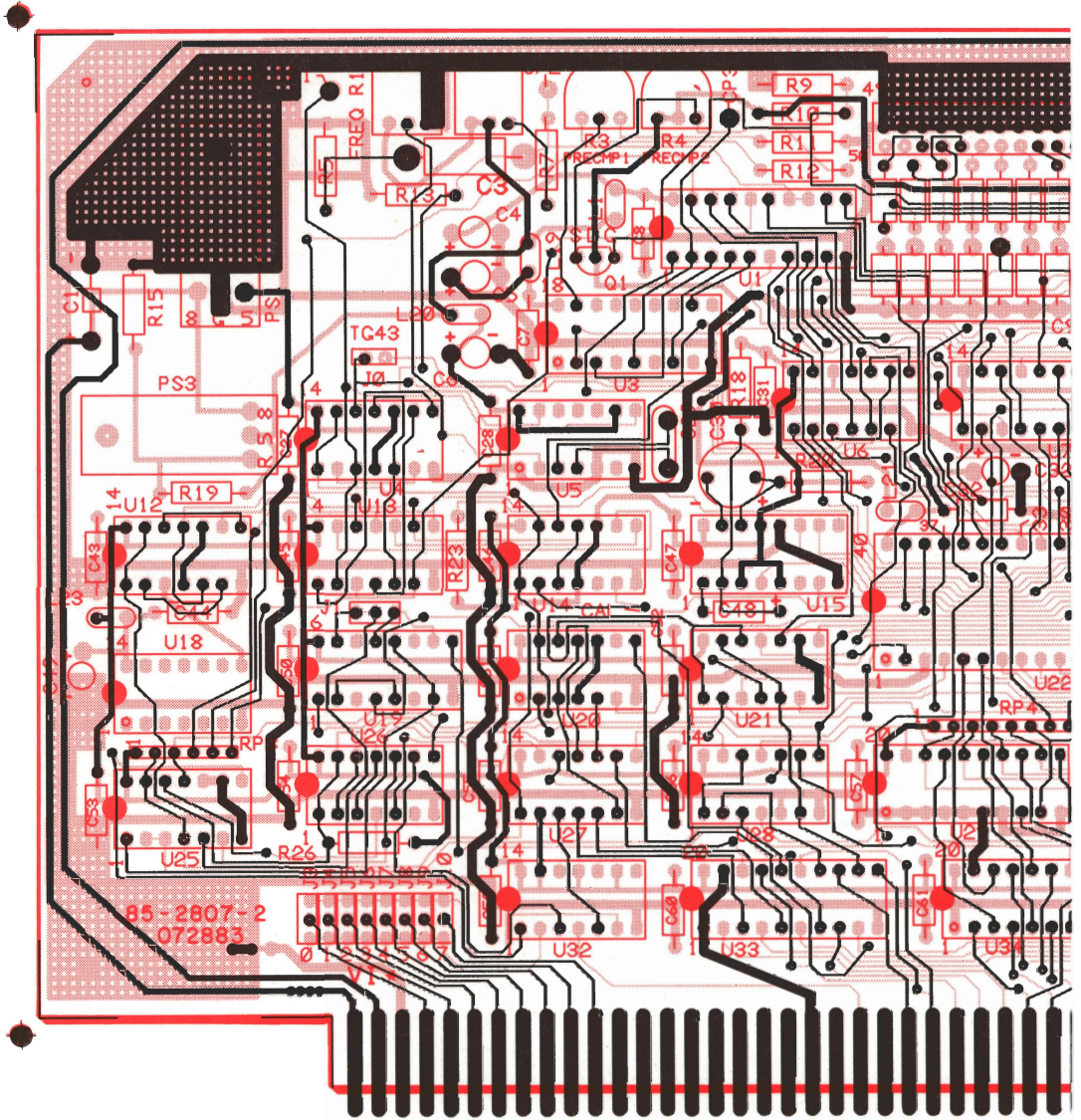
HEATH PART NUMBER	MAY BE REPLACED WITH	DESCRIPTION	LEAD CONFIGURATION (TOP VIEW)
443-1040	96LS02	MULTIVIBRATOR	
443-1063	74LS33	QUAD 2-INPUT BUFFER	

## CIRCUIT BOARD X-RAY VIEW

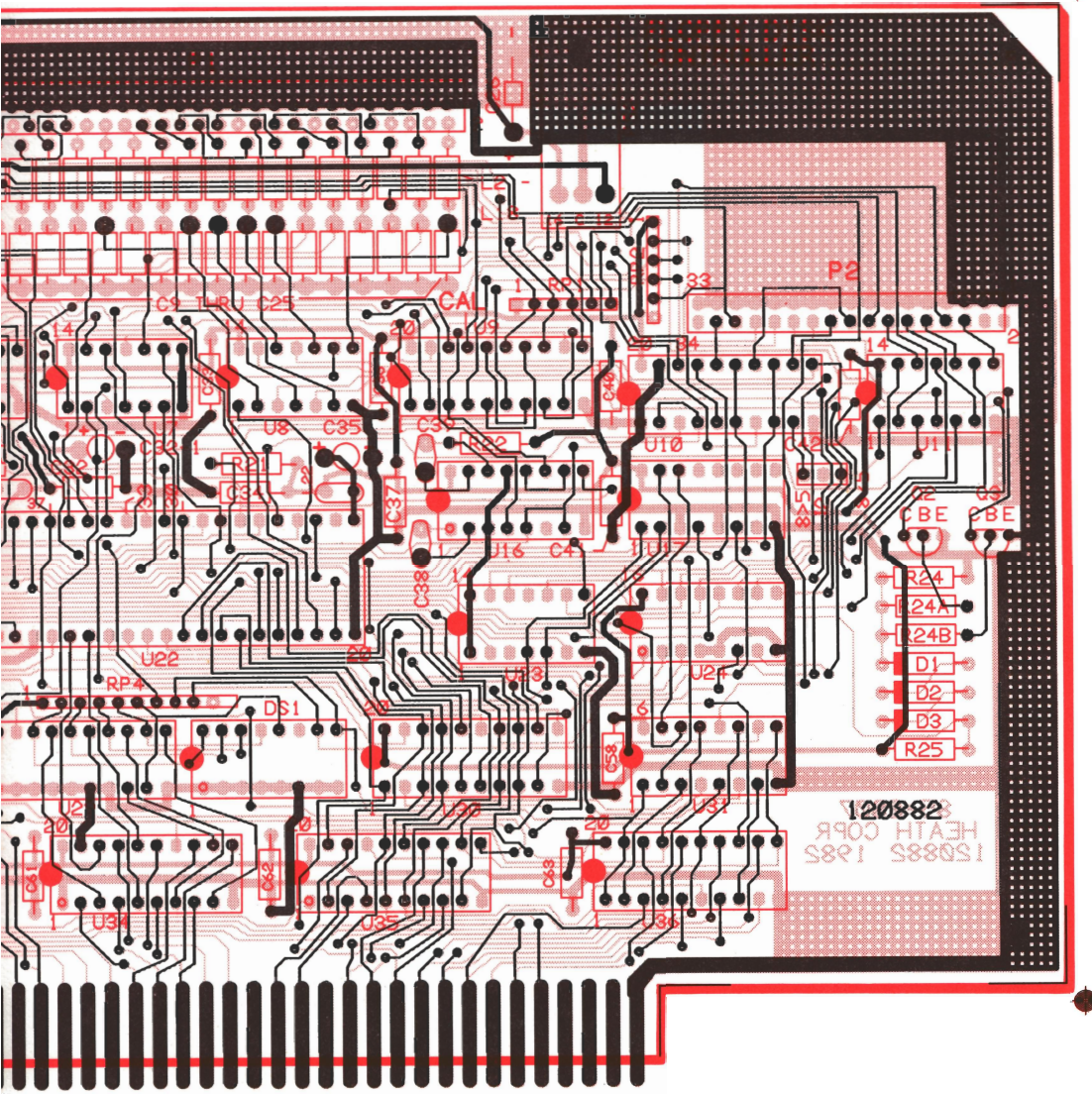
---

**NOTE:** To find the **PART NUMBER** of a component for the purpose of ordering a replacement part:

- A. Find the circuit component number (R13, R14, etc.) on the X-Ray View.
- B. Locate this same number in the "Circuit Component Number" column of the "Replacement Parts List."
- C. Adjacent to the circuit component number, you will find the **PART NUMBER** and **DESCRIPTION** which must be supplied when you order a replacement part.



**CIRCUIT I**  
**(Shown from the c**  
**component**



**CIRCUIT BOARD X-RAY VIEW**  
(from the component side. The foil on the  
component side is shown in red.)

## INTERCONNECT PIN AND SIGNAL DEFINITIONS

---

Refer to the Schematic and pages 6.13-6.15 for pin and signal numbers.

A0-A7	Address bits.
ALE	Address latch enable. Data and address lines from the CPU have valid information.
BDSEL	The H/Z-207 board is selected (enabled).
CLK	Clock signal.
CS	Chip select. When asserted, the 1797 chip is enabled.
D0-D7	Data bits on the H/Z-207 board's internal data bus.
DDEN	Double-density enable.
DI0-DI7	Data-in bits on the S-100 bus ("in" with respect to the CPU, not the controller).
DIR	Direction of drive head. When high, the drive head is stepping in. When low, the drive head is stepping out.
DO0-DO7	Data-out bits on the S-100 bus ("out" with respect to the CPU, not the Controller).
DRQ	Data request. The 1797 data register needs data for write operations or the register has data for read operations.
DSA	Drive select A. In combination with DSB, addresses the drives.
DSB	Drive select B. In combination with DSA, addresses the drives.
EARLY	Write data bit early to disk drive (used for precompensation).

## INTERCONNECT PIN AND SIGNAL DEFINITIONS

---

HLD	Head load.
HLT	Head load timing. The drive head is not engaged when this signal is low.
INDEX	The index hole on the diskette has been detected.
INTRQ	Interrupt request. H/Z-207 board has input for the CPU.
LATE	Write data bit late for drive precompensation.
MR	Master reset pin on the 1797 Controller chip that sets all registers in the chip to a known state.
pDBIN	Data request on data-in bus.
pSTVAL*	Status valid.
pSYNC	New bus cycle may begin.
PD	Pump down. Decreases the frequency of the raw read data tracking clock.
PRECOMP	Enables precompensation when low.
PU	Pump up. Increases frequency of the raw read data tracking clock.
pWR	Valid data is on data-out bus (write bus).
RAW READ	Unprocessed data from the drive.
RCLK	Clock that separates data from drive data and clock stream.



## INTERCONNECT PIN AND SIGNAL DEFINITIONS

---

RDD	Data and clock stream from the drive.
RDME	Data or status signals input for the bus are enabled.
RDY	Slave board is ready. (The H/Z-207 board is a slave board.)
RE	Read enable. Enables the 1797 chip for read operations when low.
READY	The 8" disk drive is ready.
RESET	Reset signal.
SIDE1	Otherwise known as side select output. When high, side 1 is selected in the drive. When low, side 0 is selected.
sINP	Status signal signifying data input to the bus (read cycle) may occur.
sOUT	Status signal signifying data output from the bus (write cycle) may occur.
STEP	Steps the drive head one step per pulse.
STB	Strobe output from the 1691.
TG43	Track greater than 43. The drive read/write head is over or past track 43 (track of mandatory precompensation in double-density 8" diskettes).
TK0	Track 0. The drive read/write head is over track 0 on the diskette.
TWOSIDED	The 8" drive is set for two-sided operation with a two-sided diskette.

## INTERCONNECT PIN AND SIGNAL DEFINITIONS

---

VFOE/WF	VFO enable/write fault. When WG is asserted, VFOE/WF flags write faults when deasserted, terminating any write command. When WG is deasserted, VFOE/WF enables the data separator in the 1691.
VI0-VI7*	Vector interrupts.
WAIT	RDY line is low (not ready).
WAITEN	Wait enable. Set the RDY line low on all accesses of the 1797 data register.
WD	Write data. Contains the data to be written onto the diskettes as well as the clock signals.
WDIN	Write data into the 1691 phase lock loop control.
WDOUT	Write data out of the 1691 phase lock loop and precompensation controller.
WG	Write gate. Output to the disk drive is valid.
WE	Write enable. Enables the 1797 chip for write operations.
WPRT	Write protect. When this signal is received, no write command can take place and the write protect bit in the status register is set.

## INTERCONNECT PIN AND SIGNAL DEFINITIONS

---

WRDATA	Precompensated write data pulses that have been reshaped by U16.
5DS0-5DS3	Five-inch drive select signals.
5"FASTSTEP	Enables fast stepping in the 5.25" drives.
8"/5"	Selects between the 8" and the 5.25" drives.
8DS0-8DS3	Eight-inch drive select signals.
CLOCK	Master clock signal.
01-04	Precompensation phase signals.

---

## 5-1/4" Floppy Drives

Description .....	7.2
Programming .....	7.3
Cable Connections .....	7.5
Operation .....	7.6

## DESCRIPTION

---

The Z-207-3 5-1/4" Floppy Drive is a mass storage device that stores programs and information for your computer.

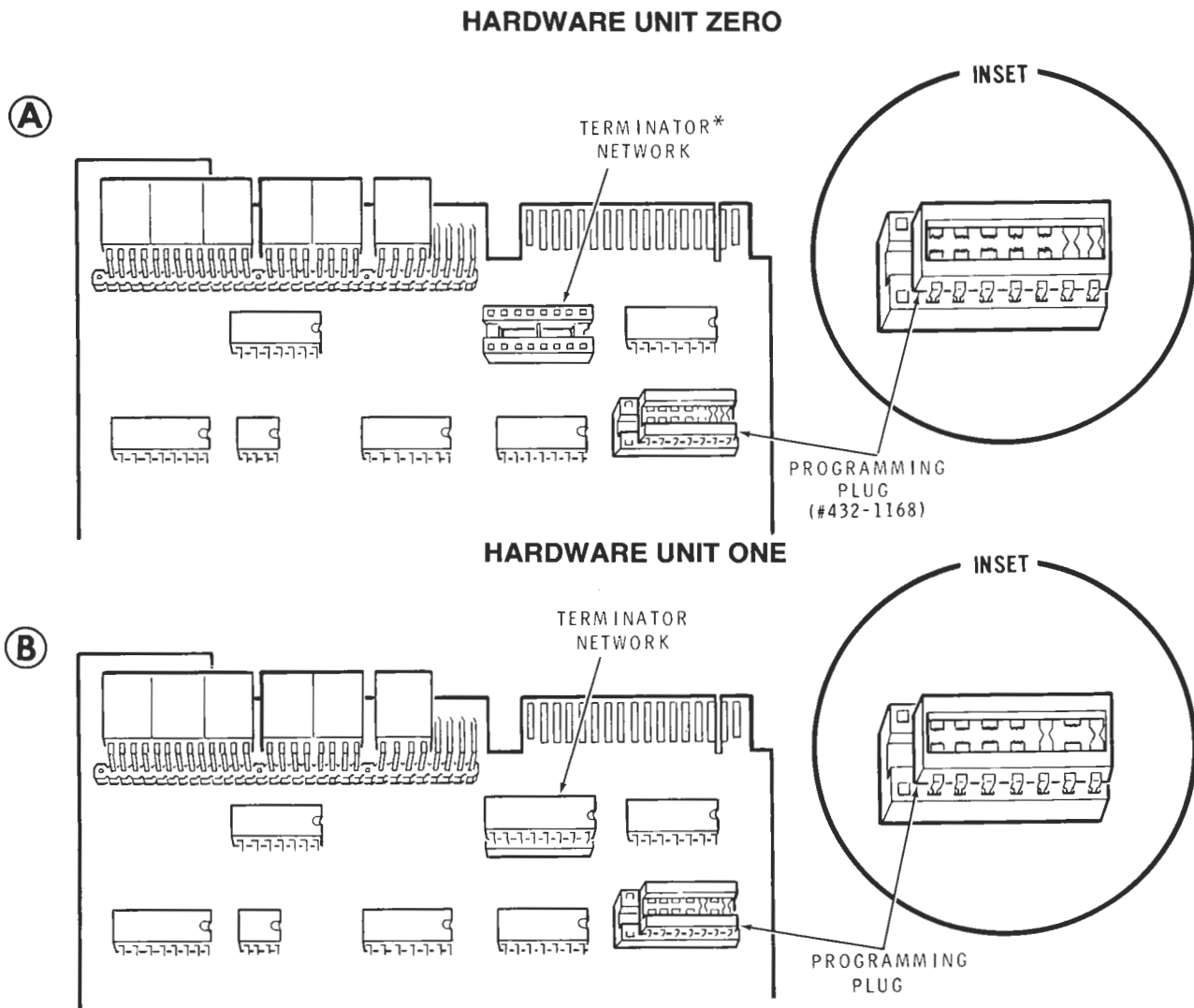
Information is stored on two sides of a 5.25-inch, oxide-coated diskette with 40 tracks per side. This drive is capable of double-density operation when it is used with a double-density controller, like the one supplied in the H/Z-100 family computers.

The recording heads are single Read/Write gap-type heads. The head carriage is positioned by a stepper motor that moves the head carriage in .02083" steps, producing 48 tracks per inch (TPI). The disk controller card in your Computer is the interface between the computer bus and the Disk Drive.

A transducer in the Drive detects the presence or absence of a notch in the diskette to insure write protection. If the notch is not detected, a signal is transmitted to the controller to indicate a read-only condition. If the notch is detected, the signal indicates a read/write condition.

The diskettes load quickly and easily through the slot in the front panel.

# PROGRAMMING



**Pictorial 7-1**  
Drive Programming

\*Must be installed in hardware unit zero if only one 5-1/4" floppy drive is installed in the Computer.

# PROGRAMMING

---

## Programming Plugs

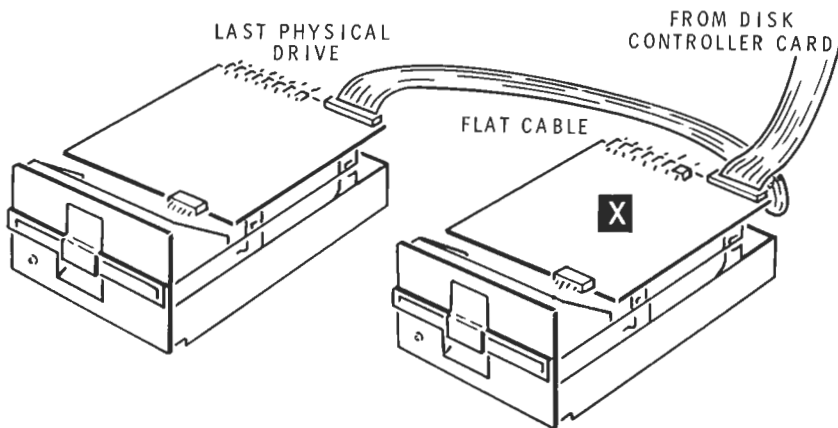
Refer to Pictorial 7-1 for the following steps.

- If this Drive is to be hardware unit 0, cut the programming plug as shown in Part A of the Pictorial.
- If this Drive is to be hardware unit 1, cut the programming plug as shown in Part B of the Pictorial.

## Terminator IC's

Each Drive is supplied with a terminator IC installed in it. (See Pictorial 7-1.) However, each Computer system, no matter how many 5-1/4" floppy drives it has, should have only one drive with a terminator IC installed in it. This terminator IC should be located in the drive that is physically last on the flat cable. Perform the following step that pertains to your system.

- If your system has only one Drive, leave the terminator IC installed in the Drive.
- If your system has two Drives, refer to Pictorial 7-2 and remove the terminator IC from Drive X.

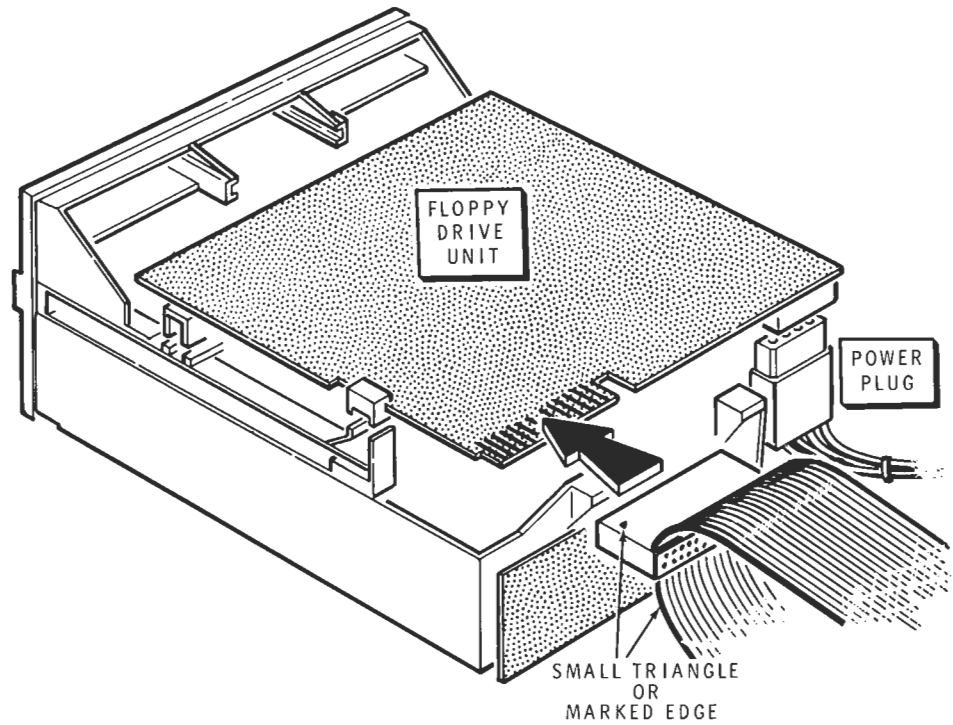


**Pictorial 7-2**  
Two-Drive System Termination

# CABLE CONNECTIONS

---

Refer to Pictorial 7-3 for a view of cable connections.



**Pictorial 7-3**  
Connecting Drive Cables



# OPERATION

---

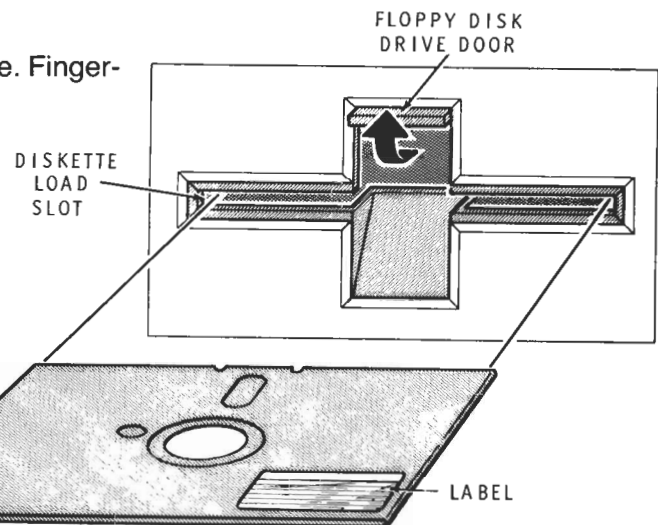
## Diskette Loading

Refer to Pictorial 7-4, open the front panel door, and insert the diskette with the label up as shown. Then close the door.

## Diskette Handling

The diskette can be easily damaged. Handle it carefully as follows:

1. Keep the diskette in its storage envelope whenever it is not in the Floppy Disk drive.
2. Keep the diskette away from magnetic fields. Magnetic fields can distort the recorded data on the diskette.
3. Replace damaged or worn storage envelopes.
4. Write on the plastic jacket only with a felt-tip pen. Do not use a lead pencil or ball-point pen.
5. Keep the diskette away from hot or contaminating materials.
6. Do not expose the diskette to sunlight.
7. Do not touch or clean the surface of the diskette. Fingerprints and abrasions can alter stored data.

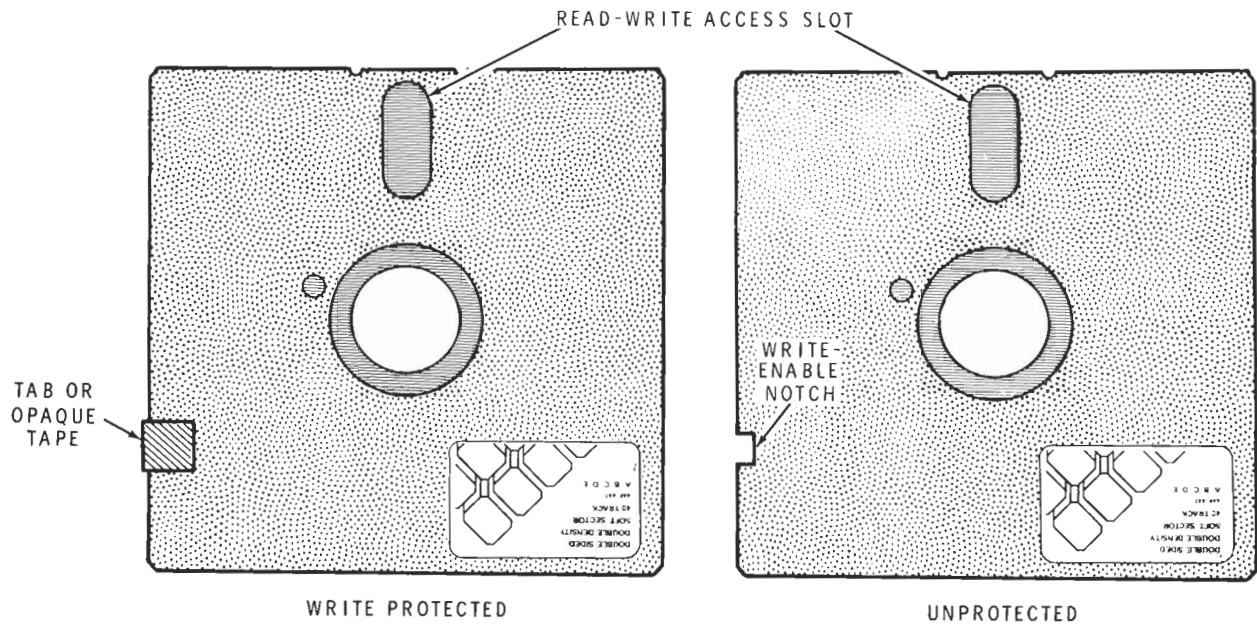


**Pictorial 7-4**  
Diskette Loading

# OPERATION

## Write-Protect

This diskette can be write protected so that it cannot be written on. To do this, cover the side notch with a tab or opaque tape. See Pictorial 7-5.



**Pictorial 7-5**  
Write Protection



# Power Supply

Power Line Considerations .....	8.2
Specifications .....	8.3

## POWER LINE CONSIDERATIONS

---

The power supply is a line-operated, voltage-fed, half-bridge, switching-type power supply. It first converts the AC line voltage to direct current and then chops this DC into a quasi-squarewave. This squarewave drives the primary of an inverter transformer. The secondary currents are converted to low voltage DC by rectifiers and filters.

The 115/230 switch (located on the rear of your Computer), is normally set at 115. This corresponds to the normal line voltage in the U.S.A. However, if you intend to use your Computer on 220 volts, reset the switch to the 230 position. (NOTE: Do not attempt to change the fuse that is inside your power supply. It is the proper value for both 115 and 230-volt operation.) Also, read and comply with the following information.

The plug on the power cord is for standard 115 VAC outlets. For 230 VAC operation in the U.S.A., replace the line cord and connector in a manner such that your power connection conforms with section 210-21 (b) of the National Electric Code, which reads, in part:

“Receptacles connected to circuits having different voltages, frequencies, or types of current (AC or DC) on the same premises shall be of such design that attachment plugs used on such circuits are not interchangeable.”

**When you install the new plug, make sure it is connected according to your local electrical code. Units with three-wire line cords must always have the green wire connected to chassis ground.**

NOTE: The power supply section of your Computer is not considered to be field serviceable. Therefore, if it ever becomes defective, you should exchange it or return it to an authorized service center.

## SPECIFICATIONS

---

AC Input Voltage .....	100 — 130 VAC, 60 Hz; 200 — 260 VAC, 50 Hz; switch selectable
Temperature Range .....	10 degrees C to 50 degrees C.
Hold Up Time .....	16 milliseconds at full load.
Current Limiting .....	130% of maximum output shuts down power supply.
Maximum Turn-on Surge .....	60 amperes for 1/2 cycle.
Oversoltage Protection .....	130% oversoltage on +5-volt line shuts down power supply.
DC Outputs .....	+ 5 VDC $\pm$ 3% at 12 amperes maximum. Including ripple, 2 amperes minimum. Ripple: 100 mV peak-to-peak maximum.
	+ 12 VDC $\pm$ 5% at 5.2 amperes maximum with + 5 VDC load at 6 amperes. Including ripple, 0.4 amperes minimum. Ripple: 120 mV peak-to-peak maximum.
	+ 8 VDC, +10%, - 5% at 8 amperes maximum. Including ripple, 150 mA minimum. Ripple: 120 mV peak-to-peak maximum.
	+ 16 VDC, +20%, - 10% at 1 ampere maximum. Including ripple, 5 mA minimum. Ripple: 150 mV peak-to-peak maximum.
	- 16 VDC, +20%, - 10% at 1 ampere maximum at 5 mA minimum. Ripple 120 mV peak-to-peak maximum.

## SPECIFICATIONS

---

All-In-One Version Only . . . . . Additional + 12 VDC  $\pm$ 5% output at 1.5 amperes maximum.  
Ripple: 50 mV peak-to-peak maximum.

Zenith Data Systems reserves the right to discontinue products and to change specifications at any time without incurring any obligation to incorporate new features in products previously sold.

## Chassis, Cabinet, & Cables

Replacement Parts List .....	9.2
Cables Location/Description .....	9.12
Circuit Boards & Hardware .....	9.17



# REPLACEMENT PARTS LIST

---

This Replacement Parts List includes the Z-100 All-in-One model and the Z-100 Low Profile model.

- Exploded Views
- Cables
- Hardware
- Circuit Boards
- Circuit Board Parts

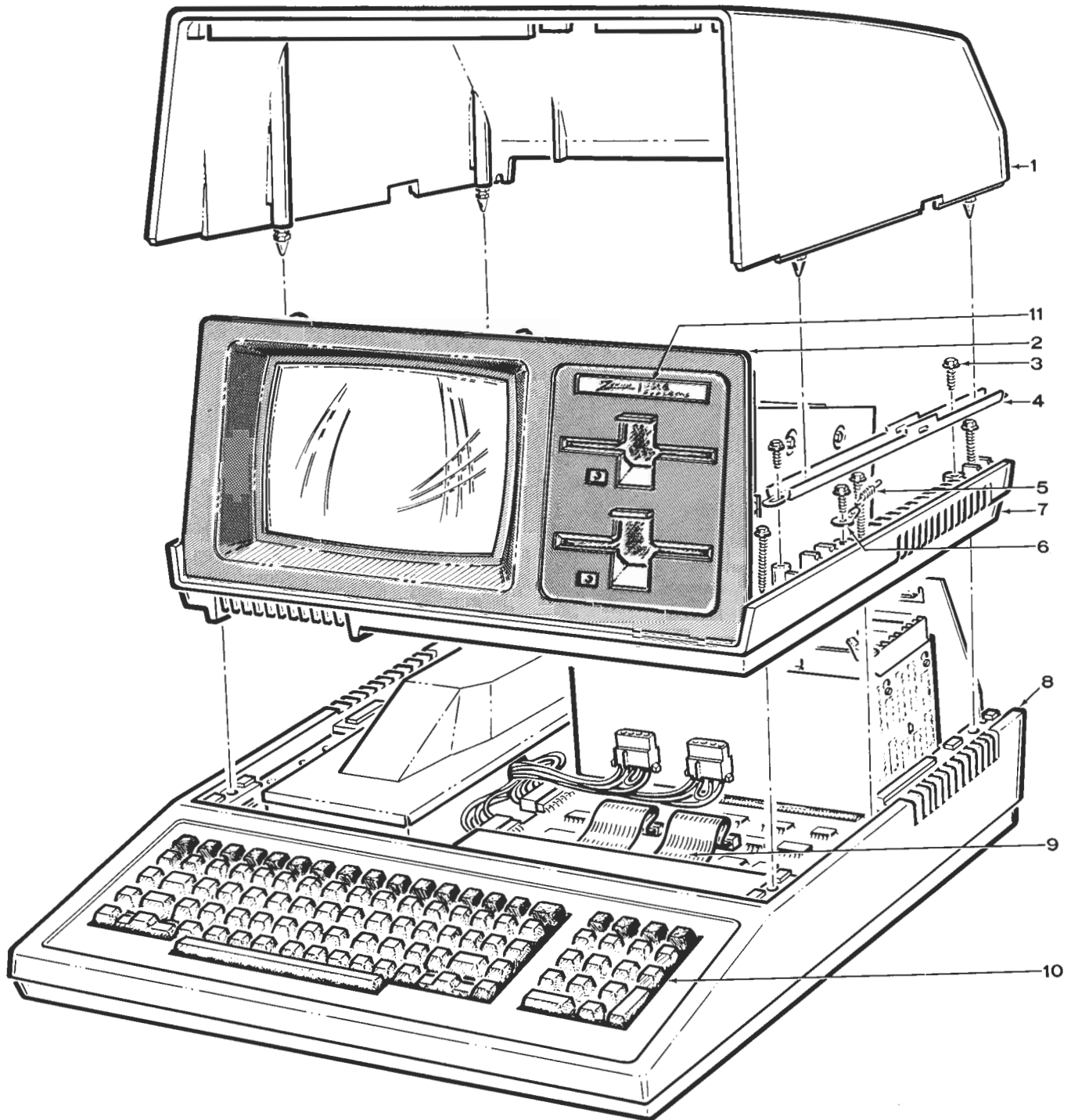
Refer to the Parts List that corresponds to your Computer (Low-Profile or All-In-One).

## All-In-One Model

The following Key Numbers correspond to the numbers on the All-in One parts pictorials. "ns" indicates a part that is not shown.

KEY NO.	PART NO.	DESCRIPTION
1	92-761	Top cover
2	92-763	Cabinet front
3	250-512	#8 self-tapping screw
4	204-2632	Slide latch
5	258-749	Spring
6	259-1	Solder lug
7	92-762	CRT drive assembly base
8	92-759	Main base
9	134-1257	40-conductor cable
10	64-899	Keyboard
11	391-653	Nameplate
ns	261-29	Rubber foot

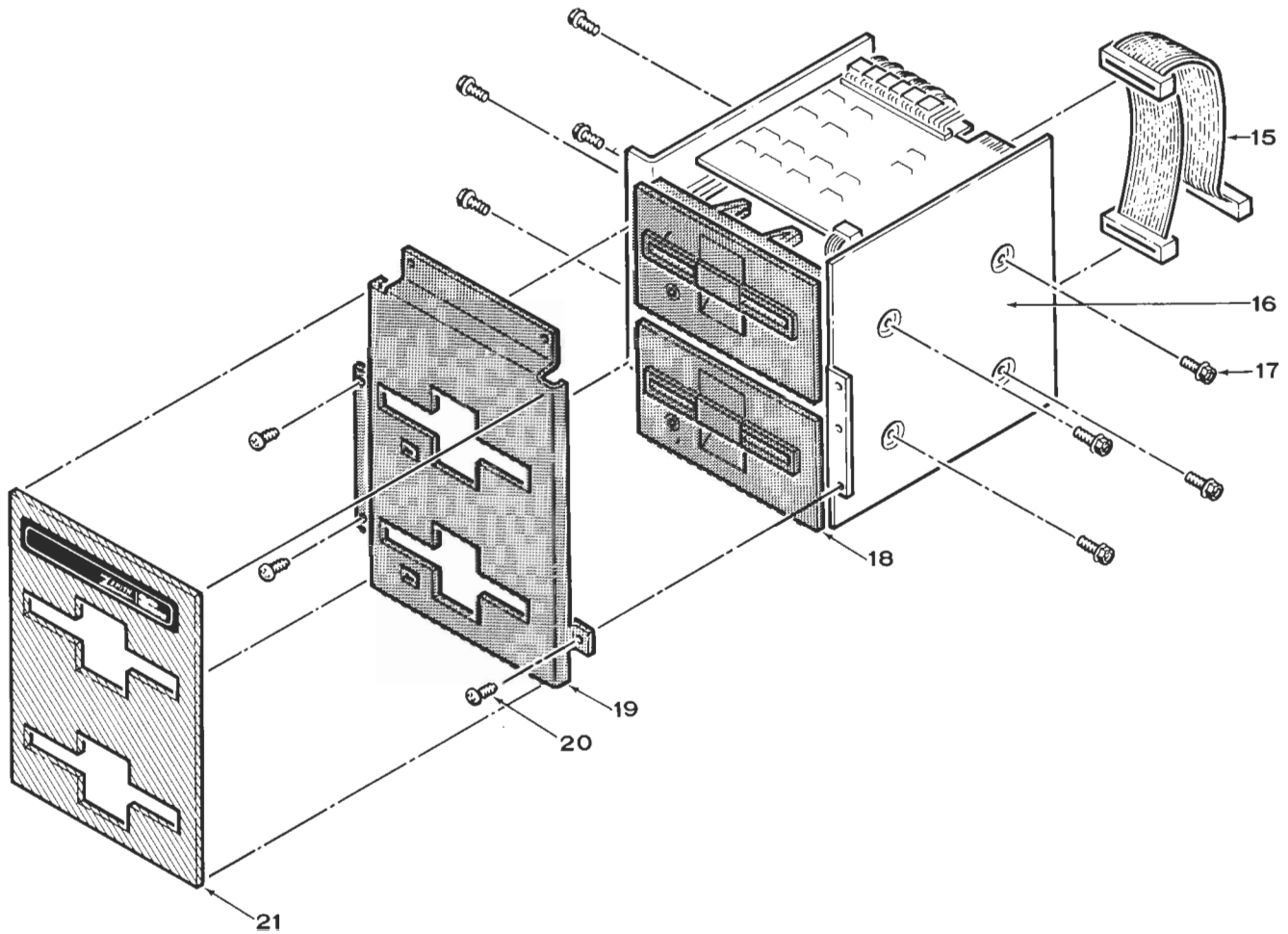
# REPLACEMENT PARTS LIST



ALL-IN-ONE

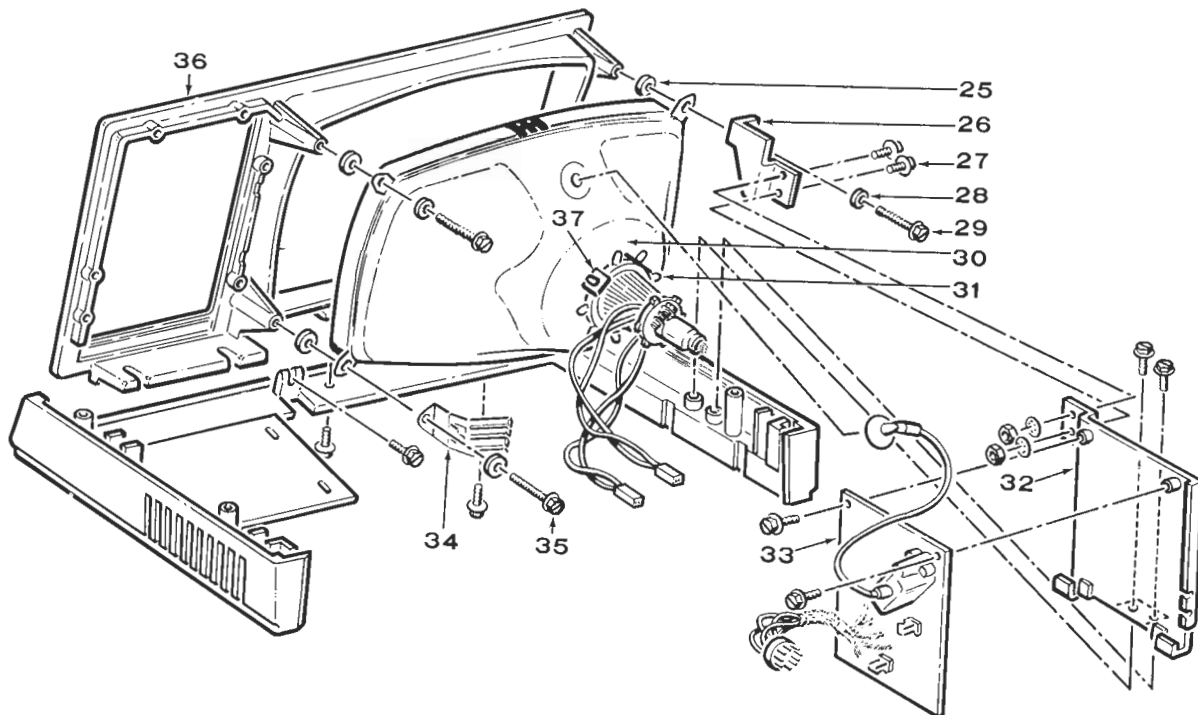
# REPLACEMENT PARTS LIST

KEY NO.	PART NO.	DESCRIPTION
15	134-1247	34-conductor cable (disk drive to disk controller board)
16	200-1419	Drive chassis mounting plate
17	250-1264	6-32 × 3/8" hex head screw
18	150-142	Disk drive (5 1/4" 48 tpi)
19	203-2129	Drive panel
20	250-1307	#6 × 1/4" phillips head screw
21	203-2131	Dual drive escutcheon
ns	203-2141	Single drive escutcheon



## REPLACEMENT PARTS LIST

KEY NO.	PART NO.	DESCRIPTION
25	73-6	Grommet
26	204-2606	CRT support bracket
27	250-1264	6-32 × 3/8" hex head screw
28	250-1318	#10 × 1-1/2" hex head screw
29	253-98	#10 flat washer
30	234-297	CRT 12" green phosphor
	234-296	CRT 12" white phosphor
	234-295	CRT 12" amber phosphor
31	234-291	Yoke
32	203-2116	CRT support
33	173-964	Video deflection board
ns	345-1	CRT ground strap
35	250-1318	#10 × 1-1/2" hex head screw
36	92-763	Cabinet front
34	234-292	Ground spring
37	234-268	Pincushion correction magnets

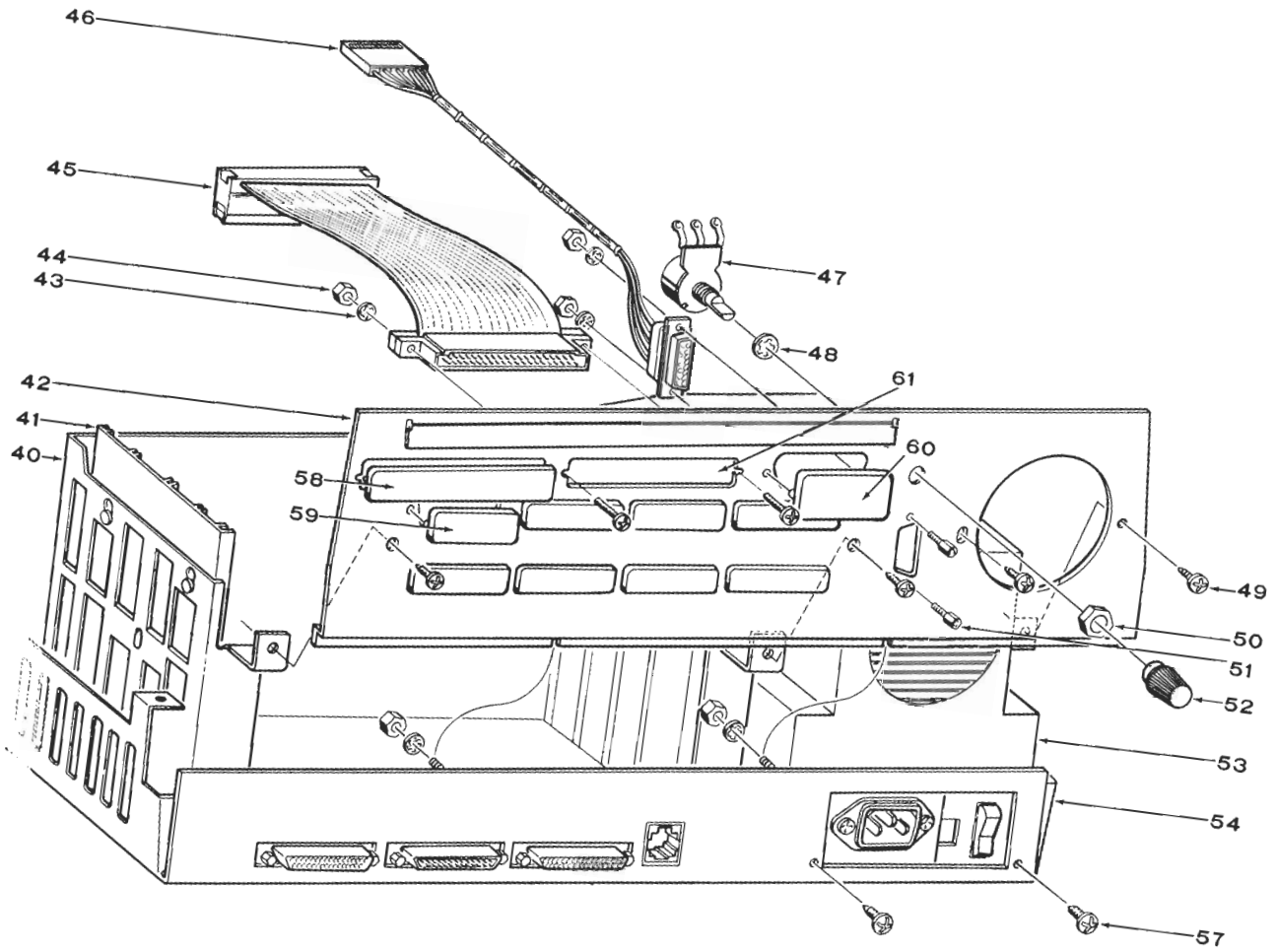


# REPLACEMENT PARTS LIST

---

KEY NO.	PART NO.	DESCRIPTION
40	206-1416	S-100 card cage
41	94-631	S-100 card rack
42	203-2139-1	Back panel
43	254-9	#4 lockwasher
44	252-2	Large 4-40 nut
45	134-1330	Floppy cable — 8" drive — 50 conductor
46	134-1254	Cable — RGB out
47	10-1192	Control — 500 $\Omega$
48	254-14	1/4" lockwasher
49	250-1307	#6 $\times$ 1/4" sheet metal screws
50	252-39	1/4" $\times$ 32 nut
51	255-757	Spacer
52	462-952	Knob
53	234-201	Power supply — All-in-One model
	234-256	All-in-One model with Winchester drive
54	200-1218-1	Chassis
57	250-1307	#6 $\times$ 1/4" sheet metal screw
58	485-44	Long plug
59	485-42	Small plug
60	485-43	Medium plug
61	485-51	Plug
ns	89-60	Line cord

# REPLACEMENT PARTS LIST



# REPLACEMENT PARTS LIST

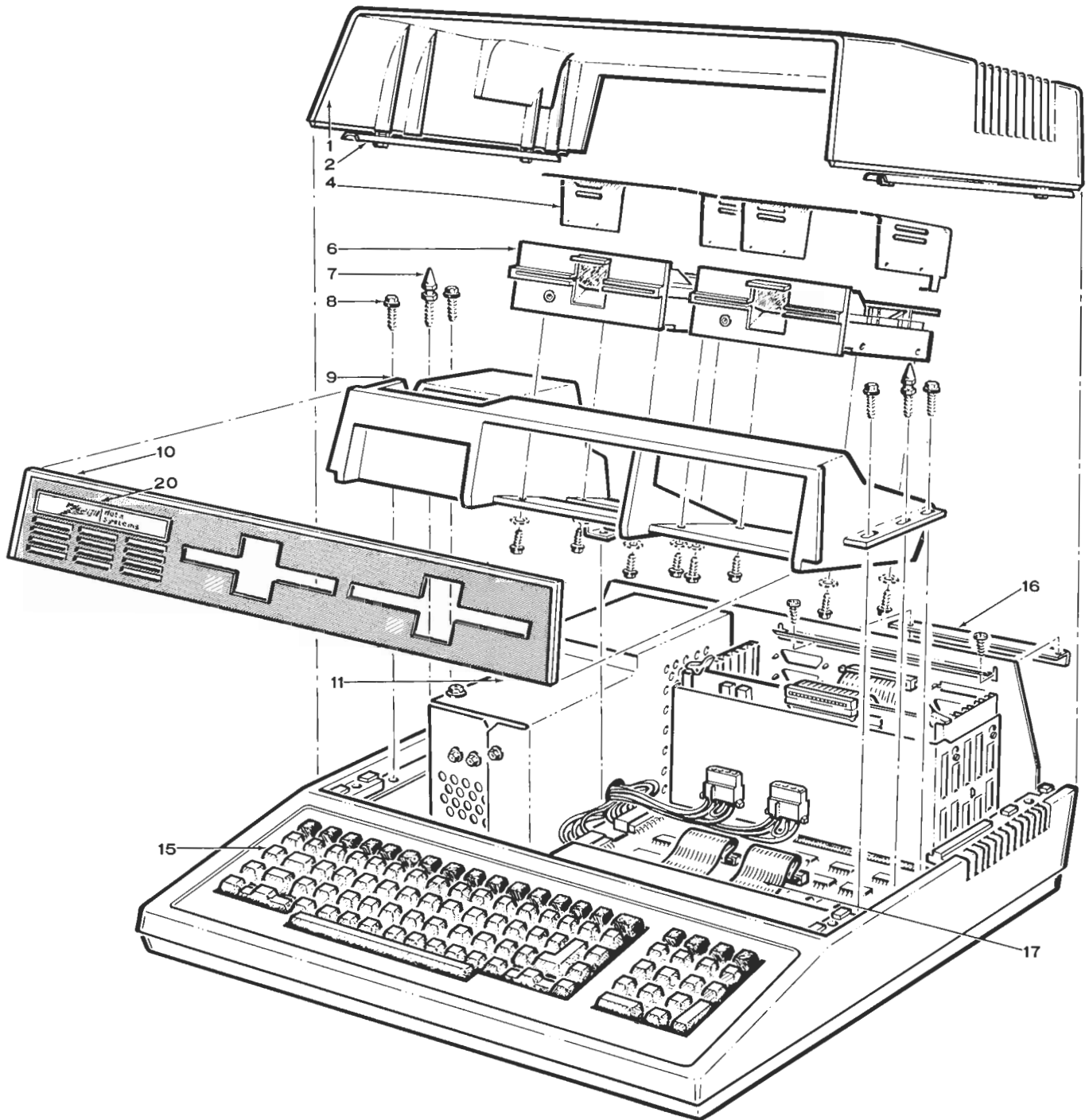
---

## Low Profile Model

The following Key Numbers correspond to the numbers on the Parts Pictorials. ns indicates part not shown.

KEY PART NO.	NO.	DESCRIPTION
1	92-758	Top cover
2	204-2605	Slide rail
ns	258-750	Spring
4	206-1456	Drive shield
6	150-142	Disk drive 5-1/4", 48 tpi
7	262-56	Threaded pin
8	250-512	#8 × 3/4" self-tapping screw
9	92-760	Drive shelf
10	203-2125	Escutcheon – dual drive
ns	203-2124	Escutcheon – single drive
11	234-200	Power supply – Low Profile
	234-257	Power supply – Lo Profile model with Winchester drive
15	64-899	Keyboard
16	204-2638-1	Cable clamp
17	134-1257	40-conductor cable – video logic board to main board
ns	255-804	Large spacer to support video logic board
ns	261-29	Rubber foot
20	391-658	Nameplate

# REPLACEMENT PARTS LIST



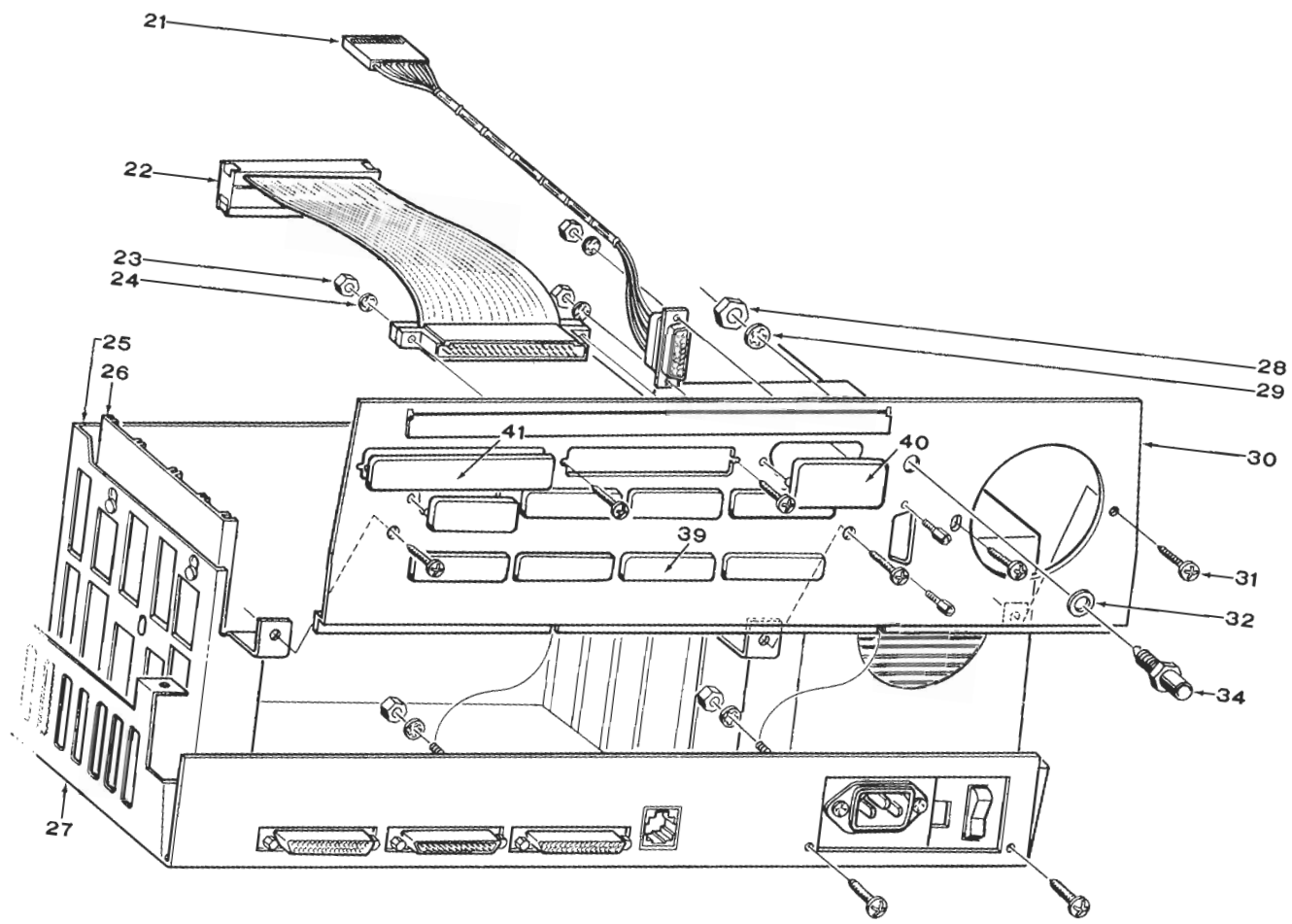


# REPLACEMENT PARTS LIST

---

KEY NO.	PART NO.	DESCRIPTION
21	134-1254	7-conductor video RGB cable
22	134-1330	50-conductor flat cable
23	252-2	Large 4-40 nut
24	254-9	#4 lockwasher
25	94-631	Card rack
26	206-1416	Card cage
27	200-1418-1	Chassis
28		Nut
29	254-6	#6 washer
30	203-2139-1	Back panel
31	250-1307	#6 × 1/4" sheet metal screw
32	254-14	1/4" lockwasher
34	434-107	Phono socket
38	485-42	Small plug
39	485-43	Medium plug
40	485-44	Large plug
ns	89-60	Line cord

# REPLACEMENT PARTS LIST



## CABLES LOCATION/DESCRIPTION

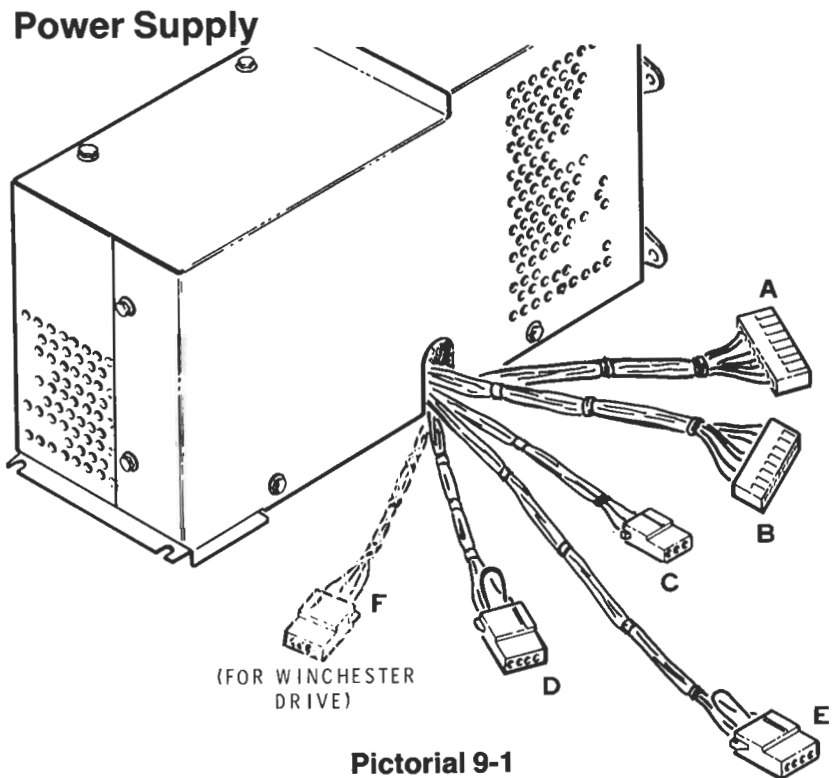
---

### CABLES

<u>PART NO.</u>	<u>DESCRIPTION</u>
134-1330	50-conductor flat cable. From J16 on the rear panel to P1 on the disk controller board.
134-1257	40-conductor flat cable. From P304 and P305 on the video logic board to P104 and P106 on the main board.
134-1246	34-conductor flat cable. From J1 of each disk drive to P2 on the disk controller board.
134-1254	7-wire cable. From J9 on the rear panel to P303 on the video deflection board.
134-1265	Shielded cable from J14 on the rear panel to P301 on the video logic board.
89-60	Power line cord
89-65	Power line cord – Class B units

## CABLES LOCATION/DESCRIPTION

The following lists provide you with a description and location of the cables and connectors used in your Z-100 Low-Profile or All-In-One Computer. Part numbers for these cables are listed in the Replacement Parts List in this manual.



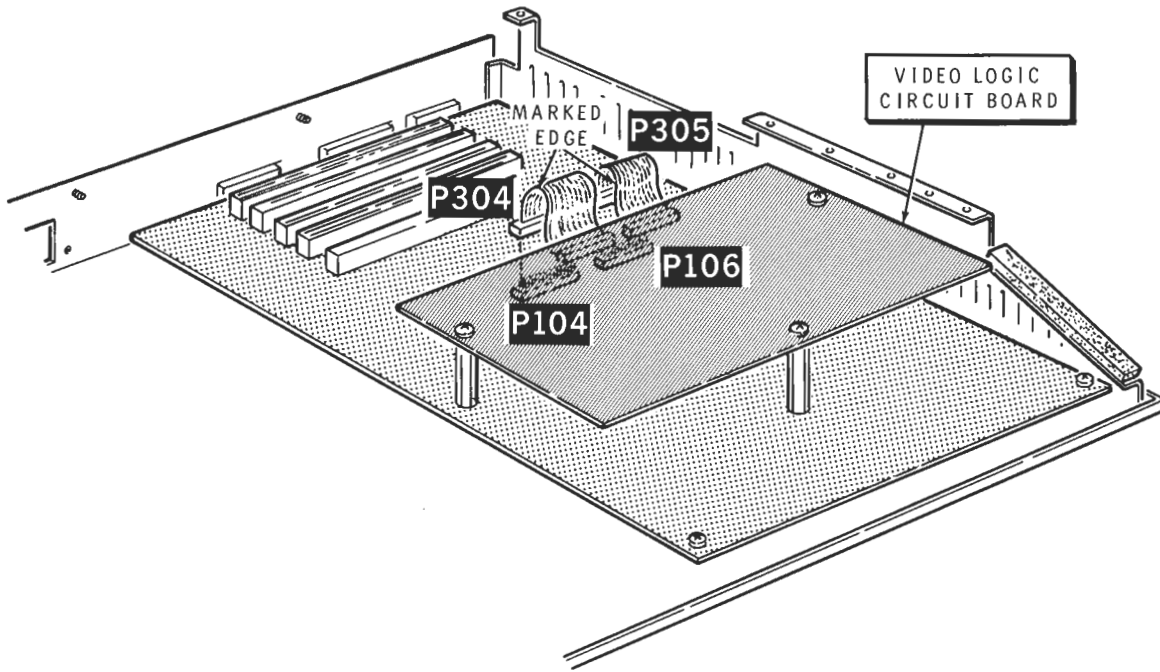
**Pictorial 9-1**  
Power Supply Cables

- A 8-wire socket (blk,wht,blu,blu,blu,yel,blk) to main board plug P101
- B 8-wire socket (blk,blk,red,red,red,blk,blk) to main board plug P102
- C 4-wire socket (red,blk,blk,wht) to the data separator card
- D 4-wire socket (red,blk,blk,org) to the disk drive. Plug P4 on your power supply may have two ferrite beads with sleeving on the red wire.
- E 4-wire socket (red,blk,blk,org) to the disk drive. Plug P5 on your power supply may have two ferrite beads with sleeving on the red wire.
- F 4-wire socket (red,red,blk,blk) to the Winchester drive

# CABLES LOCATION/DESCRIPTION

---

## Video Logic Board



**Pictorial 9-2**  
Video Logic Cables

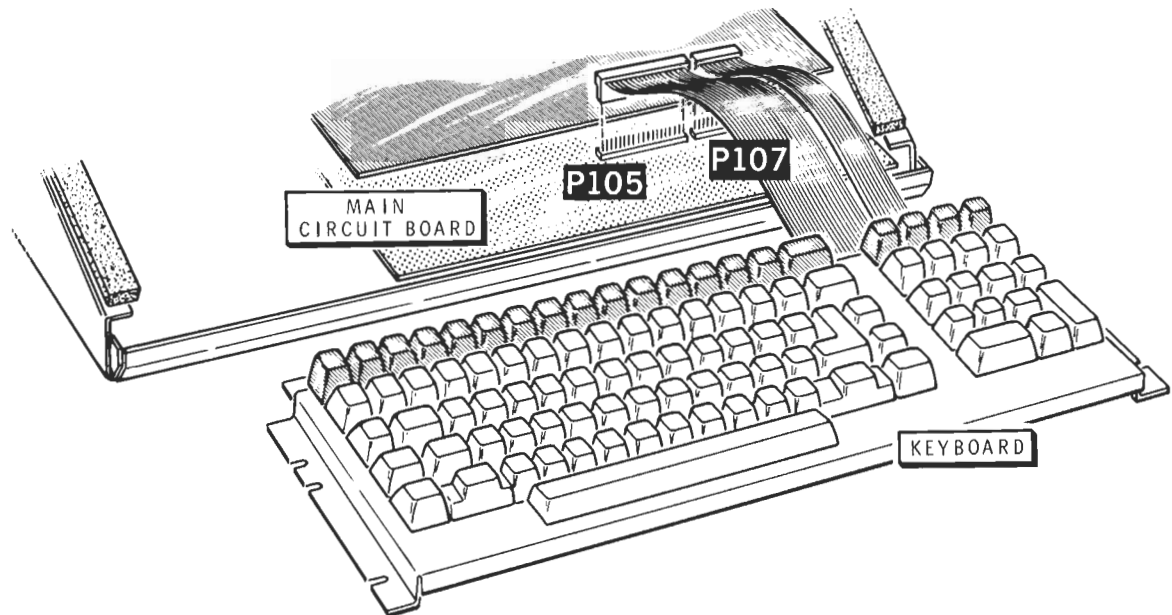
40-conductor cable from plug P104 to the main board plug P304. Part number 134-1257.

40-conductor cable from plug P106 to the main board plug P305. Part number 134-1257.

## CABLES LOCATION/DESCRIPTION

---

### Keyboard



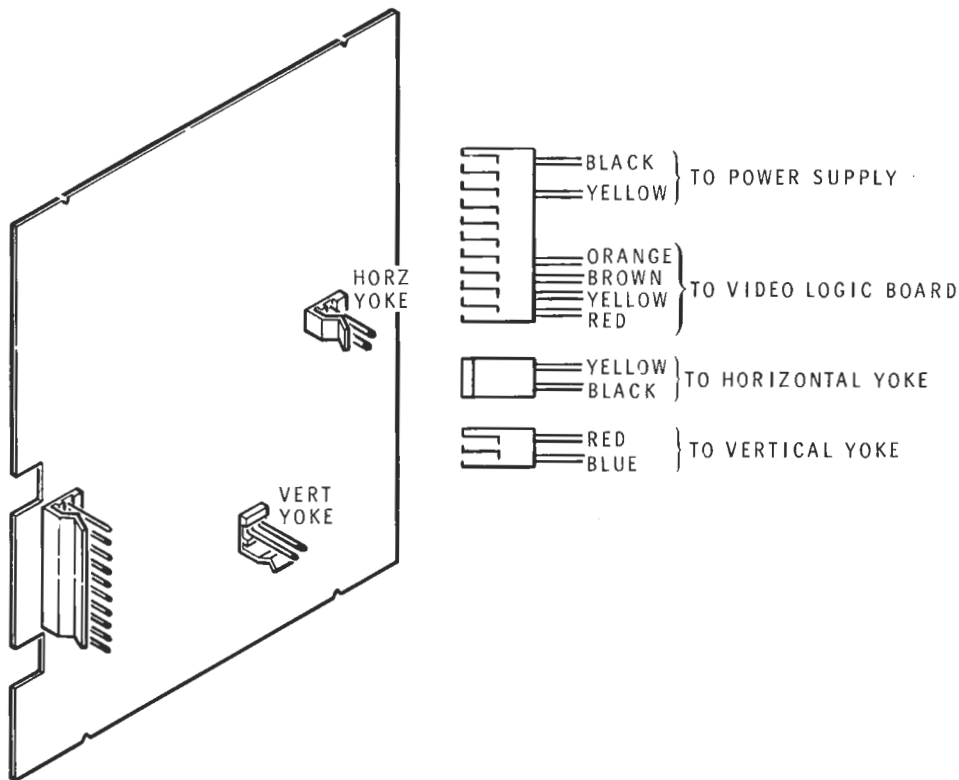
**Pictorial 9-3**  
**Keyboard Cables**

20-conductor flex cable to the main board plug P105  
10-conductor flex cable to the main board plug P107

# CABLES LOCATION/DESCRIPTION

---

## Video Deflection Board



**Pictorial 9-4**  
**Video Deflection Cables**

6-wire socket from the power supply and video board to the 10-pin plug on the video deflection board

2-wire socket from the horizontal yoke to the video deflection board (horiz yoke plug)

2-wire socket from the vertical yoke to the video deflection board (vert yoke plug)

# CIRCUIT BOARDS & HARDWARE

---

## Circuit Boards

<u>PART NO.</u>	<u>DESCRIPTION</u>
181-3630	Main board (8K ROM)
181-4106	Main board (16K ROM)
181-3631	Video logic board (B/W)
181-3267	Video logic board (color)
181-3763	Floppy disk controller board
234-202	Video deflection board

## Hardware

<u>PART NO.</u>	<u>DESCRIPTION</u>
-----------------	--------------------

### #4 Hardware

250-1411	4-40 × 1/4" screw
250-1413	4-40 × 1/2" screw
254-9	#4 lockwasher
252-15	Small 4-40 nut
252-2	Large 4-40 nut

### #6 Hardware

250-1422	6-32 × 1/4" flat head screw
250-1307	#6 × 1/4" sheet metal screw
250-1325	6-32 × 1/4" pan head screw
250-1264	6-32 × 3/8" hex head screw
250-1199	#6 × 5/8" self-tapping screw
254-6	#6 external lockwasher
254-1	#6 internal lockwasher
252-3	6-32 nut

### Other Hardware

262-56	Threaded pins
250-512	#8 × 3/4" self-tapping screw
255-757	Small spacer
255-804	Large spacer
258-750	Spring





---

# Programming Data

Description .....	10.2
General Information .....	10.3
Devices Permitting User Programming .....	10.10
Port Addresses .....	10.11
Z-DOS Initialization Sequence .....	10.14
ASCII Chart .....	10.30
Escape Codes .....	10.38
Escape Codes Defined .....	10.42
Key Code Chart .....	10.52
Keypad Code Chart .....	10.59
Function Key Code Chart .....	10.60

## DESCRIPTION

---

This section of the Manual provides condensed system programming information. It is provided for the experienced programmer to help him understand the Computer System so he can develop his own software or firmware.

## GENERAL INFORMATION

---

### 8085 Key Facts

Clock Speed:	5 MHz.
Address Space:	16 bits extended to 24.
Interrupts:	TRAP = (NMI or power failure) RST5.5, RST6.5, RST7.5 disabled. Vectored interrupts through 8259 disabled by Mask.
DMA:	External devices, and processor swap.
Reset:	Keyboard function.

### 8088 Key Facts

Clock Speed:	5 MHz.
Address Space:	20 bits, extended to 24.
Interrupts:	NMI = (NMI or power failure). TEST jumperable to 0 or 1. Vectored by 8259.
DMA:	External devices, and processor swap.
Reset:	Keyboard function.

### ROM Information

Size:	4, 8, 16, or 32 kilobytes (by jumpers).
Address:	1016 – 1023K or, top 8k or every 64k or, every 8k in memory or, can be deselected.

### RAM Information

Size:	64k to 192k, in 64k increments, parity standard.
Address:	Dependent on mapping ROM.

## GENERAL INFORMATION

---

### Interrupt structure

Device Type: 8259A.  
 Number: Master standard; slave optional.

<u>LEVELS</u>	<u>MASTER</u>
0 (Highest)	Error = Parity error or (S-100 pin 98).
1	Processor swap interrupt.
2	Timer (8253 Out 0 or Out 2).
3	Slave 8259.
4	Serial port A.
5	Serial port B.
6	Keyboard or display.
7	Printer.

<u>LEVELS</u>	<u>SLAVE</u>
0	S-100 vectored interrupt 0.
1	" " " 1.
2	" " " 2.
3	" " " 3.
4	" " " 4.
5	" " " 5.
6	" " " 6.
7	" " " 7.

## GENERAL INFORMATION

---

### Processor Swapping

The Computer contains two processors, with selection circuitry to enable the desired processor. Processor swap occurs when the presently selected processor writes to bit 7 (MSB) of the processor swap port (PSP). A 1 selects the 8088 and a 0 selects the 8085. The processor swap port is port FE.

When a processor swap occurs, the newly selected processor can restart from where it left off, or an interrupt can keep it from starting. Interrupt generation is enabled by writing a 1 to bit 1 of the PSP.

If interrupts are not masked, the currently selected processor is signalled when an interrupt is requested. If the MASK mode is selected, no interrupts will get through to the 8085 and the 8088 will service all interrupts. In the MASK mode, the 8088 is selected whenever an interrupt occurs. MASK is bit 0 (LSB) of the PSP. A 1 activates this function.

<u>Bit</u>	<u>Definition</u>
0	0 = Both processors receive interrupts 1 = Force a processor swap to 8088 if 8085 active
1	0 = Resume execution from previous address 1 = Force an interrupt as newly selected processor becomes active
7	0 = Select 8085 processor 1 = Select 8088 processor

## GENERAL INFORMATION

---

### Memory Mapping

Four options affect how the ROM is addressed. These options are enabled when the program writes to the memory control latch (MEMCTL). Latch bits 2 and 3 control the four options (bit 3 = MSB). The MEMCTL latch is at port FC.

The first option is used for power up. The two control bits are zero (B3=0, B2=0). In this mode, the ROM appears to be in all of address space during reads. Memory writes occur normally. The ROM code will perform a far jump (into itself), and then select another ROM addressing option.

The following chart shows which port bits control the four **ROM** configurations.

<u>BITS</u>	<u>DEFINITION</u>	
3,2	00 = Option 0	01 = Option 1
	10 = Option 2	11 = Option 3

Option 0, the power-up or master reset configuration, makes the code in ROM appear to be in all of memory when reads are performed. Writes, however, occur normally.

Option 1 makes the ROM code appear to be at the top of every 64 K page of memory.

Option 2 makes the ROM code appear to be at the top of the first megabyte of memory.

Option 3 disables the ROM.

When the ROM is selected, all other memory (except video RAM) is deselected to allow other memory to "share" the ROM's address space (phantom). NOTE: Be careful not to select video RAM when option 0 or 1 is enabled, and be careful when you select the video RAM when option 1 is selected.

## GENERAL INFORMATION

---

RAM normally consists of from one to three banks of 64K bytes. This provides from 64 to 192K bytes of memory.

The RAM address configuration depends on the map control bits in MEMCTL. Bit 0 is MAPSEL0 and bit 1 is MAPSEL1.

The following chart shows which port bits control the various **RAM** configuration.

<u>BITS</u>	<u>DEFINITION</u>
1,0	00 = Option 0      01 = Option 1 10 = Option 2      11 = Option 3

Option 0, the power-up and master reset configuration, provides contiguous addressing; from 0 to 192K.

Option 1 swaps the RAM block from 0 to 48K with the block at 64 to 112K.

Option 2 swaps the RAM block from 0 to 48K with the block at 112 to 160K.

Option 3 swaps the RAM block from 4 to 60K with the block at 68 to 124K.

### Parity

Parity consists of a parity bit for each byte in RAM. This adds one, two, or three 64K-bit chips (depending on how much RAM is installed: 64K, 128K, or 192K) and the associated support circuitry.

RAM parity has two control options: ZERO\_PARITY and KILL \_PARITY. The ZERO\_PARITY option sets parity to the zero state regardless of the data pattern that was written, and forces a parity error to check the parity logic. The option is activated when the system writes a 0 to bit 4 of the Memory Control Latch (MEMCTL) port.



## GENERAL INFORMATION

---

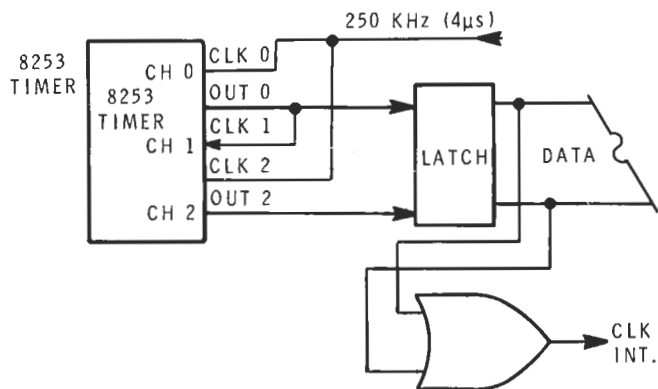
The KILL \_PARITY option disables the parity checking circuitry. This option is enabled when the system writes a 0 to bit 5 of the MEMCTL port. It also clears a parity error by first writing a 0 to bit 5 of the port and then a 1 to bit 5 of the port.

### Timer

The 8253 timer has three channels. (See Pictorial 10-1). Each channel has an input (CLK), and an output (OUT). As shown, channels 0 and 1 are cascaded. CLK0 and CLK2 are tied to a 250 kHz (4  $\mu$ s) clock, and the CLK1 input is tied to the output of channel 0.

The two outputs that are available externally are OUT0 and OUT2. These are ORed together to produce the timer interrupt input of the 8259. A latch is provided which, when read by software, determines which of the channels caused the interrupt (TMRSTAT).

TMRSTAT must be cleared by the program after it is read. Bit 0 of TMRSTAT corresponds to OUT0, and bit 1 is OUT2. The appropriate latch is cleared by writing a 0 to that bit of TMRSTAT.



**Pictorial 10-1**

## GENERAL INFORMATION

---

The 8253 data sheet is supplied in the Appendices portion of this documentation. The following chart is provided for the convenience of those who may already be familiar with the 8253 device.

<u>BIT</u>	<u>DEFINITION</u>	
0	0 = Use 16-bit binary counter 1 = Use 4-decade binary coded decimal counter	
1	000 = Mode 0	001 = Mode 1
2	X10 = Mode 2*	X11 = Mode 3
3	100 = Mode 4	101 = Mode 5
4	00 = Counter latch	01 = Read/load least significant byte
5	10 = Read/load most significant byte	11 = Read/load least significant byte, then most significant byte
6	00 = Counter 0	01 = Counter 1
7	10 = Counter 2	11 = undefined

\*X = Don't care.

## DEVICES PERMITTING USER PROGRAMMING

---

Several of the major IC's in your Computer are user programmable. Please refer to the manufacturer's data sheets in the Appendices portion of this documentation for programming information. These IC's include:

8259's	Interrupt controllers
6845	CRT controller (CRT-C)
2661's	Synchronous/asynchronous data communications controller
6821's	Parallel interface controller
8253	Timer
1797	Floppy disk controller

Also included in the Appendices is the S-100 proposed specifications, the 8085 instruction set and the IAPX 88 Book, which includes the 8088 instruction set.

## PORT ADDRESSES

---

The following chart lists the input/output port assignments for the H/Z-100 series computers.

<u>Device</u>	<u>Port Address (in hexadecimal)</u>
DIP Switch SW101	0FF
Processor Swap Port	0FE
High Address Latch	0FD
Memory Control Latch	0FC
8253 Timer Status	0FB
reserved by ZDS	0F6-0FA
8041A Keyboard Processor	0F4-0F5
8259A Master Interrupt Controller	0F2-0F3
8259A Slave Interrupt Controller	0F0-0F1
2661 Serial B (Modem Port)	0EC-0EF
2661 Serial A (Printer Port)	0E8-0EB
8253 Timer	0E4-0E7
Parallel Port (Main Board)	0E0-0E3
reserved by ZDS	0DF
Light Pen Control	0DE
6845 CRT Controller	0DC-0DD
Video 68A21 Parallel Port	0D8-0DB
reserved by ZDS	0C0-0D7
ET-100 Trainer Parallel Input/Output	0D4-0D7*
ET-100 CRT Controller	0CD-0CE*
Secondary Floppy Disk Controller	0B8-0BF
Primary Floppy Disk Controller	0B0-0B7
Primary Winchester Controller	0AE-0AF
Secondary Winchester Controller	0AC-0AD
reserved by ZDS	0A8-0AB

\* The ET-100 cannot house any S-100 cards. Therefore, future optional cards may use these addresses.

## PORT ADDRESSES

---

<u>Device</u>	<u>Port Address (in hexadecimal)</u>
Gateway (reserved)	0A4-0A7
Network Card (NET-100)	0A0-0A3
Expansion Memory Boards (Z-205)	098-09F
reserved by ZDS	084-097
Development Port (Temporary)	080-083
Primary Multiport Card (Z-204)	060-07F
Secondary Multiport Card (Z-204)	040-05F
reserved for non-ZDS vendors	000-03F

## Memory Assignments

<u>Device</u>	<u>Port Address (in hexadecimal)</u>
MTR-100	
(Monitor ROM — Firmware)	0F000:0C000-0FFFF
reserved by ZDS	0F000:01000-0BFFF
Network Card (NET-100)	0F000:00000-00FFF
Video RAM (Green Plane)	0E000:00000-0FFFF
Video RAM (Red Plane)	0D000:00000-0FFFF
Video RAM (Blue Plane)	0C000:00000-0FFFF
User RAM	00000:00000-0B000:0FFFF
ET-100 Reserved Addresses (in addition to those listed above)	
MTRET-100	
(Monitor ROM — Firmware II)	0F000:08000-0BFFF
MTRET-100	
(Monitor ROM — Firmware I)	0F000:04000-07FFF

Since the ET-100 trainer cannot accommodate S-100 cards, future H/Z-100 S-100 cards may utilize these memory locations.

## PORT ADDRESSES

---

### Parallel Port

The parallel port is designed around U114 (68A21), the Peripheral Interface Adapter. The IC performs three functions: It operates as a printer port, it serves as a port for the light pen and it couples the video board vertical retrace signal to the CPU. The CPU accesses the PIA for programming or data transfer. At the same time it will chip-select the PIA by asserting the 6821CS control line from the I/O port decoder. The CPU asserts the  $\overline{\text{OUT}}$  line, pin 21, when the Computer needs to write to the PIA. In all other cases, the PIA will remain in the read mode. Data transfer takes place when the CPU asserts  $\overline{\text{W0}}$  for a write or  $\overline{\text{DBIN}}$  for a read.

### Light Pen

The light pen circuits consists of four ICs. By itself, the CPU will not respond to a signal from the light pen circuits. It requires a user-supplied program to set up interrupts, handle timing, and take care of bit locations pointed to by the light pen.

## Z-DOS INITIALIZATION SEQUENCE

---

This section describes all phases of Z-DOS initialization from the time that control is passed from the system ROM until Z-DOS gives control to COMMAND.COM for standard system operation. Following the initialization description are some sample initialization programs.

Z-DOS start-up is basically a two-step process. First, the loader is loaded from sector zero of the diskette. Then, the loader loads IO.SYS from the diskette and passes control to IO.SYS. The IO.SYS:

- Loads the operating system.
- Loads COMMAND.COM.
- Passes control to COMMAND.COM.

The following sections thoroughly discuss each of these functions.

### The Loader

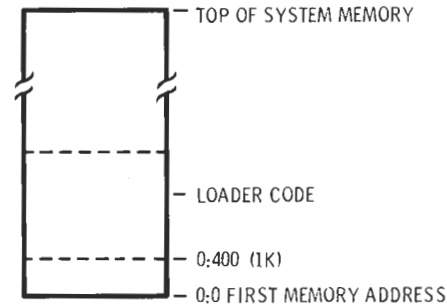
The purpose of the loader is to load in IO.SYS and pass control to it. The first 512 bytes of every diskette (or full sector if sector sizes are greater than 512 bytes) is reserved for the loader. The loader resides in a known location on the diskette (the first 512 bytes) so that the system ROM can correctly locate it when the user issues the BOOT command to the ROM.

The loader is placed on the diskette by the FORMAT program. The same loader is used on each type of diskette, whether it is 8 inch or 5-1/4 inch. A small table located at offset 3 in the loader contains specific information about the type of diskette that it resides on. This disk information is placed in the loader by FORMAT just before FORMAT writes the loader onto a newly formatted diskette.

## Z-DOS INITIALIZATION SEQUENCE

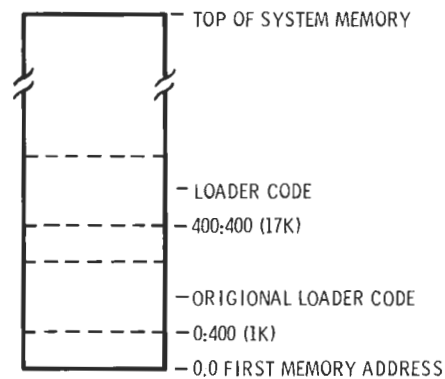
---

The system ROM loads in the loader at address 0:400. The contents of the instruction pointer (IP) is therefore 400, and the code segment (CS) register is 0. The other registers are assumed to contain random data. The loader assumes that the system ROM read is enough of track zero so as to have at least the first sector of the director already in RAM. This may be anywhere from 4 sectors to 17 sectors, depending on the disk format. This allows the loader to be smaller, since it does not have to read in the directory. (See Pictorial 10-2.)



**Pictorial 10-2**  
Loader

The first thing the loader does is to relocate itself. This is because it must load IO.SYS at address 40:0. The loader relocates itself to address 400:400 and performs a long jump to this address + current IP. (See Pictorial 10-3.)



**Pictorial 10-3**  
Relocation of Loader



## Z-DOS INITIALIZATION SEQUENCE

---

Secondly, the loader sets up the registers into the 8080 memory model (CS = DS = SS = ES) and proceeds to collect information passed to it by the system ROM. This information includes boot device number, port address, and boot string. It then locates the ROM to the top of the 8088 address space.

Next, the loader locates the proper address that contains the first sector of the directory, and insures that the first named file is IO.SYS. It also determines if the diskette drive should be double stepped, which is necessary if 48 tpi media is used in a 96 tpi drive. Once IO.SYS is located, its starting sector number and size in sectors is computed, and it is then read into memory at address 40:0. Note that the loader assumes that IO.SYS file is contiguous on the diskette and is less than 16 K bytes long.

The disk layout of track 0 is:

- Track 0
- Sector 0
  - Loader (512 bytes).
  - FAT\* #1 (Varies).
  - FAT #2 (Same as FAT #1, used for backup).
  - Directory (Varies, 32 bytes per entry).
  - Data space to end of diskette.

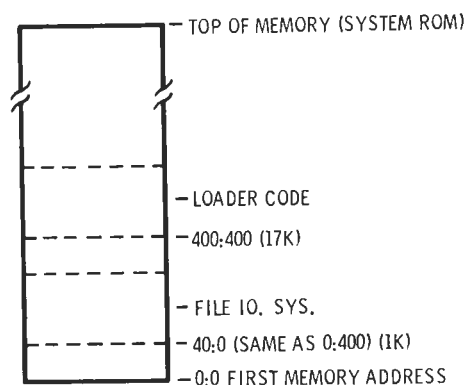
\*FAT (File Allocation Table)

## Z-DOS INITIALIZATION SEQUENCE

The specific disk layouts for the different diskette formats is as follows. The first number is the starting sector number, and the number in parenthesis is the size in sectors.

	<u>48 ss</u>	<u>48 ds</u>	<u>96</u>	<u>8" ss</u>	<u>8" ds</u>
Loader	0 (1)	0 (1)	0 (1)	0 (4)	0 (1)
FAT #1	1 (1)	1 (1)	1 (1)	4 (6)	1 (2)
FAT #2	2 (1)	2 (1)	2 (1)	10 (6)	3 (2)
Directory	3 (4)	3 (7)	3 (9)	16 (26)	5 (6)
Data	7 (313)	10 (630)	12 (1268)	42 (1960)	11 (1221)
Bytes/sector	512	512	512	128	1024

Once IO.SYS has been read into memory at address 40:0 (see Pictorial 10.4), the loader executes a far jump to IO.SYS, and the source index (SI) register points to the diskette parameter table mentioned in the discussion of FORMAT above. The only error messages issued by the loader will be "No System" if IO.SYS is not the first file on the diskette, or "I/O error" if a read error occurs while IO.SYS is being loaded. Note that the loader does not do retries on read operations.



**Pictorial 10-4**  
**Loader-IO.SYS**

## Z-DOS INITIALIZATION SEQUENCE

---

### IO.SYS

IO.SYS is entered at address 40:0, with SI pointing to the disk parameter table contained in the loader (see the following table). The IO.SYS insures an 8080 memory model (CS = DS = ES = SS), and then sets its stack pointer to a memory address in the IO.SYS workspace. It then moves the loader information table into a known location in IO.SYS for future access.

#### Loader Disk Parameter Table

Byte	0–2	=	Near JMP.
	3	=	Version number (should be 1).
	4–5	=	Sector size in bytes.
	6	=	Sectors per cluster.
	7	=	Number of reserved sectors.
	8–9	=	Number of FATs (should be 2).
	10–11	=	Number of director entries.
	12–13	=	Number of sectors on the disk.
	14	=	Log 2 of sector size.
	15	=	Sectors per track.
	16–17	=	First sector number of data area.
	18	=	Log 2 of cluster factor.
	19–20	=	First sector of directory area.
	21	=	Flag byte.
	Bit 0	=	1 if double-sided.
	1	=	1 if fast stepped.
	2–3	=	Not used.
	22	=	Select byte.
	Bit 0–1	=	Should be zero.
	2	=	1 if 8" drive.
	3	=	Always = 1.
	4	=	1 if to use precomp.
	5–6	=	Not used.
	7	=	1 if single-density.
	23–24	=	Port number of controller.

## Z-DOS INITIALIZATION SEQUENCE

---

IO.SYS next moves the ROM work space to the IO.SYS's workspace, so that it will not conflict with other pieces of the system.

Then the IO.SYS performs as follows:

1. The interrupt vectors are all initialized to the default interrupt handler address, the wild interrupt handler.
2. The new ROM data segment address, and the keyboard interrupt handler address are set into the interrupt page.
3. The interrupt routine addresses for the timer, slave 8259A, serial ports A and B, keyboard/display/light pen, parallel port, and the eight slave interrupt lines from the 8259A are set into the interrupt vector page.
4. The keyboard, serial A and serial B, and the PIA port are initialized with mode bytes and command port clearing, along with direction information for the PIA.
5. The light pen is set to cause CA1 to be set, but not issue interrupts on a 0 to 1 transition, and the V sync is set to cause CA2 to be set and to cause an interrupt on 0 to 1 transitions.

## Z-DOS INITIALIZATION SEQUENCE

---

6. The timer is then initialized, and a test is made to insure that the timer is functioning properly.
7. The slave 8259A is set for level-triggered cascading, and the 8086 interrupt is set to fully nested and non-buffered. The master 8259A is also set for the same configuration. At this point, interrupts are enabled.

The configuration information (setup by CONFIGUR) is now used to initialize the Z-DOS devices PRN, AUX, and CON. The defaults for these devices are:

PRN — Serial A, 4800 baud, DCD high  
AUX — Serial B, 4800 baud, DCD high  
CON — System CRT

At this point the sign-on message is printed, identifying IO.SYS, and IO.SYS and loader are checked for compatible revision numbers. If this is true, IO.SYS uses the information concerning device unit and port number, passed to it by the loader, to set up its disk tables and the default drive name used by Z-DOS.

## Z-DOS INITIALIZATION SEQUENCE

---

### Z-DOS Drive Mapping

For each 5-1/4" and 8" drive in the system, IO.SYS restores the drive head. If it sees an invalid track zero indication, it marks the drive as imaginary in the drive table. Otherwise, it issues "10 steps out" and then another restore. Once it locates all the non-existent drives, it then locates a matching existing drive, and maps the imaginary drive into that existing drive. If it finds no existing drive, it marks the imaginary drive as real, and sets the flag to indicate that the system should not allocate any imaginary drives to this drive. This is done for the user who forgets to power on his 8" drives at boot time. Note that the existence/non-existence of 5 1/4" drives does not affect the settings of the 8" drives, or vice versa. See the following chart.

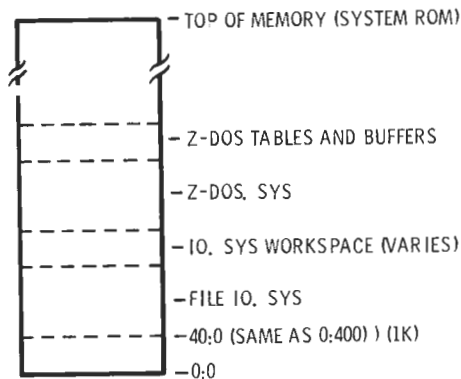
<u>Drives Not Located</u>	<u>Physical Drives</u>			
	<u>5 1/4"</u>	<u>5 1/4"</u>	<u>8"</u>	<u>8"</u>
A:	I		R	
B:		R	I	
A: & B:	R	R		
C:			I	R
D:			R	I
C: & D:			R	R

NOTE: I indicates that the drive is imaginary, and R indicates that the drive is real. An imaginary drive will use the first real drive of the same type for all of its I/O.

## Z-DOS INITIALIZATION SEQUENCE

---

The next address for the final location for the file Z-DOS.SYS is found by adding the total size of IO.SYS and its required work areas. Then the file Z-DOS.SYS is searched for in the directory. Z-DOS.SYS must be the second name in the directory for IO.SYS to be able to locate it. Again, the Z-DOS.SYS file is assumed to be contiguous and less than 64K bytes long. During proper operation the file Z-DOS.SYS is read into memory, the DOS\_INIT routine in Z-DOS.SYS is called with pointers to the disk parameter table, and a flag tells Z-DOS to size memory. (See Pictorial 10-5.)



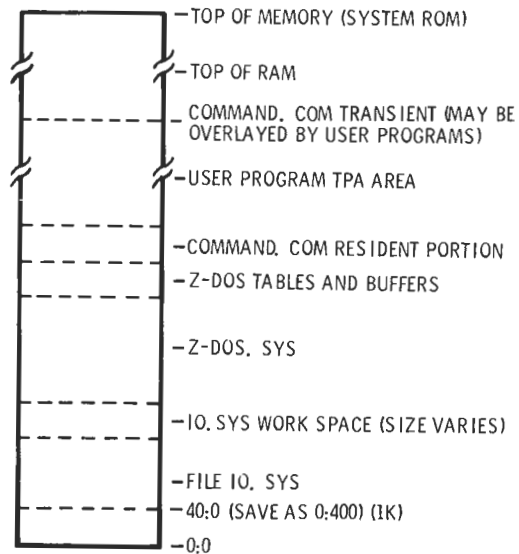
**Pictorial 10-5**  
Z-DOS.SYS

The disk parameter table defines the number of valid disk drives, as well as all the possible sector/directory/diskette capacities. DOS\_INIT uses this information to calculate disk buffers, FAT buffers, and do some preliminary memory initialization and internal table setup. DOS\_INIT is located with the file Z-DOS.SYS.

## Z-DOS INITIALIZATION SEQUENCE

---

On return from DOS\_INIT, IO.SYS turns on the keyboard, and then uses the appropriate function calls to load and execute the file COMMAND.COM. Once COMMAND.COM is loaded, control is passed to it, at which point it initializes itself, prints its header, checks for AUTOEXEC.BAT, gets the date and time from the user, and prints the system prompt. Then the operating system is fully in control, and the user is ready to begin executing programs. (See Pictorial 10-6.)



**Pictorial 10-6**  
Initialization Complete



## Z-DOS INITIALIZATION SEQUENCE

---

### **Sample Programs**

NOTE: Label definitions for the following programs can be found in the Z-DOS Distribution Disk II definitions files, or in Appendix I of Volume II of the Z-DOS Manual.

# Z-DOS INITIALIZATION SEQUENCE

---

```

;
; Initialize the Keyboard
;
        MOV     AL,ZKEYDK           ; Disable keyboard until ready
        OUT     ZKEYBRDC,AL
BINITKEY1:
        IN      AL,ZKEYBRDS        ; Wait for command to complete
        TEST    AL,ZKEYIBF
        JNZ     BINITKEY1

        MOV     AL,ZKEYCF          ; Flush typeahead buffer
        OUT     ZKEYBRDC,AL
        IN      AL,ZKEYBRDD

;
; Save video state set up by the ROM monitor
;
        IN      AL,ZVIDEO+PIADATA ; Get current video state
        MOV     BYTE PTR VIDEO_ROM,AL ; Save it

;
; Initialize the Serial port A
;
        XOR     AL,AL              ; Turn off unit
        OUT     ZSERA+EPCMD,AL
        IN      AL,ZSERA+EPCMD     ; Reset mode reg ptr
        MOV     AL,EPB1+EPCL8+EPA16X ; Set mode reg 1
        OUT     ZSERA+EPMODE,AL
        MOV     AL,EPMR2A+EPB960 ; Set mode reg 2
        OUT     ZSERA+EPMODE,AL
        MOV     AL,EPNORM+EPRTS+EPRESE+EPRXEN+EPDTR
        OUT     ZSERA+EPCMD,AL    ; Set Command port
        IN      AL,ZSERA+EPDATA   ; Clear input
        IN      AL,ZSERA+EPDATA

;
; Initialize the Serial port B
;
        XOR     AL,AL              ; Turn off unit
        OUT     ZSERB+EPCMD,AL
        IN      AL,ZSERB+EPCMD     ; Reset mode reg ptr
        MOV     AL,EPB1+EPCL8+EPA16X ; Set mode reg 1
        OUT     ZSERB+EPMODE,AL
        MOV     AL,EPMR2A+EPB960 ; Set mode reg 2
        OUT     ZSERB+EPMODE,AL
        MOV     AL,EPNORM+EPRTS+EPRESE+EPRXEN+EPDTR
        OUT     ZSERB+EPCMD,AL    ; Set Command port
        IN      AL,ZSERB+EPDATA   ; Clear input
        IN      AL,ZSERB+EPDATA

```

## Z-DOS INITIALIZATION SEQUENCE

---

```

;
; Initialize PIA port
;
      MOV     AL,PIADDAC      ; Set control ports for data
      OUT     ZPIA+PIACTLA,AL
      OUT     ZPIA+PIACTLB,AL
      MOV     AL,01011111B   ; Load initial data value for port A
      OUT     ZPIA+PIADATA,AL
      MOV     AL,11111111B   ; Load initial data value for port B
      OUT     ZPIA+PIADATB,AL
      XOR     AL,AL          ; Set control ports for direction
      OUT     ZPIA+PIACTLA,AL
      OUT     ZPIA+PIACTLB,AL
      MOV     AL,10101111B   ; Set direction reg for port A
      OUT     ZPIA+PIADDRA,AL
      MOV     AL,11111100B   ; Set direction reg for port B
      OUT     ZPIA+PIADDRB,AL

; Make 0->1 transitions of light pen cause CA1 to be set, but do not cause interrupt
; Make 0->1 transitions of vsync cause CA2 to be set and cause interrupts

      MOV     AL,PIADDAC+PIAC12+PIAC23 ; Set control port A for data
      OUT     ZPIA+PIACTLA,AL

; Disable transitions of parallel printer to cause interrupt

      MOV     AL,PIADDAC      ; Set control port B for data
      OUT     ZPIA+PIACTLB,AL

; Clear CA1,CA2 and CB1,CB2 by reading the port and then using
; dummy I/O to drive clock on PIA (and thus cause clear to occur)

      IN      AL,ZPIA+PIADATA ; Clear CA1,CA2
      IN      AL,ZPIA+PIADATB ; Clear CB1,CB2
      IN      AL,ZDIPSW       ; Read from a "safe" place(the dip switch)

; Turn off clear of light pen/vsync so CA1/CA2 transitions can occur

      IN      AL,ZPIA+PIADATA ; Get current data value
      OR      AL,10100000B    ; Turn off clear of Vsync/Light pen flipflops
      OUT     ZPIA+PIADATA,AL ; Allow vsync to cause interrupts

;
; Initialize the Timer
;
;
; Make sure all counter read cycles are completed

      IN      AL,ZTIMER+PITCO
      IN      AL,ZTIMER+PITCO

```

# Z-DOS INITIALIZATION SEQUENCE

---

```

        IN      AL,ZTIMER+PITC1
        IN      AL,ZTIMER+PITC1
        IN      AL,ZTIMER+PITC2
        IN      AL,ZTIMER+PITC2

; Init counter modes

        MOV     AL,PITSCO+PITRLW+PITMSW
        OUT    ZTIMER+PITCW,AL ; Counter 0 - square wave generator
        MOV     AL,PITSC1+PITRLW+PITMITC
        OUT    ZTIMER+PITCW,AL ; Counter 1 - event counter
        MOV     AL,PITSC2+PITRLW+PITMITC
        OUT    ZTIMER+PITCW,AL ; Counter 2 - intr on terminal count

; Init counter values

        XOR     AL,AL           ; Timer 1
        OUT    ZTIMER+PITC1,AL
        OUT    ZTIMER+PITC1,AL
        MOV     AX,ZTIMEVAL     ; Timer 0
        OUT    ZTIMER+PITC0,AL
        MOV     AL,AH
        OUT    ZTIMER+PITC0,AL

; Wait for first rising clock from counter 0

        MOV     AL,OFFH-ZTIMERS0
        OUT    ZTIMERS,AL
        XOR     CX,CX           ; Get timeout value
TIMEL:
        IN      AL,ZTIMERS      ; Get status
        TEST   AL,ZTIMERS0     ; Has it occurred yet ?
        LOOPZ  TIMEL           ; No, try again
        JNZ    BINIC1         ; If clock responded, then skip
        JMP    TIMERR         ; Clock never responded
BINIC1:

; Clear any pending interrupts

        MOV     AL,OFFH-ZTIMERS0-ZTIMERS2
        OUT    ZTIMERS,AL

;
; Initialize parity generation
;

        MOV     AL,BIOS_MCL     ; Get value of memory control latch
        TEST   AL,ZMCLPK       ; Is parity checking specified ?
        JZ     MEMIF           ; No, skip
        AND    AL,NOT ZMCLPK   ; Turn off checking(and on generation)
        OUT    ZMCL,AL        ; Output value

; Set up the Z-205 boards

        MOV     DX,Z205BA      ; DX = base address

```

# Z-DOS INITIALIZATION SEQUENCE

---

```

MEMILO:  MOV    CX,Z205BMC      ; CX = number of boards
          OUT    DX,AL          ; Set it up
          LOOP  MEMILO

MEMIL:   XOR    AX,AX          ; Start at segment 0
          MOV    DS,AX          ; Set up segment regs
          MOV    ES,AX
          MOV    AX,DS:0       ; Get first word
          MOV    BX,AX         ; Save a copy
          INC    AX             ; Bump value
          INC    WORD PTR DS:0 ; Bump memory
          CMP    AX,WORD PTR DS:0 ; Are they the same ?
          JNE    MEMIC         ; No, skip
          MOV    WORD PTR DS:0,BX ; Restore value
          XOR    SI,SI         ; Set up regs for move
          XOR    DI,DI
          MOV    CX,08000H     ; Get number of words to move
          REP  MOVSW          ; Move words onto themselves

MEMIC:   MOV    AX,DS          ; Get segment
          ADD    AX,01000H     ; Point to next segment, finished ?
          JNZ    MEMIL        ; No, try again

          MOV    AX,CS         ; Restore seg regs
          MOV    ES,AX
          MOV    DS,AX

MEMIF:

;
; Initialize the Slave 8259A interrupt controller
;
          MOV    AL,ICW10P+ICW1LT+ICW1IL4
          OUT    ZS8259A+ICW1,AL ; Level triggered, cascaded
          MOV    AL,ZS8259AI    ; Get slave base interrupt number
          OUT    ZS8259A+ICW2,AL ; Set interrupt base number
          MOV    AL,ZINTSLV
          OUT    ZS8259A+ICW3,AL ; Set slave number
          MOV    AL,ICW4UPM+ICW4SFN
          OUT    ZS8259A+ICW4,AL ; Set processor to 8088, fully nested, non buffered

          MOV    AL,OFFH
          OUT    ZS8259A+OCW1,AL ; Don't allow any interrupts

;
; Initialize the Master 8259A interrupt controller
; (interrupts are still disabled)
;
          MOV    AL,ICW10P+ICW1LT+ICW1IL4
          OUT    ZM8259A+ICW1,AL ; Level triggered, cascaded
          MOV    AL,ZM8259AI    ; Get Master base interrupt number
          OUT    ZM8259A+ICW2,AL ; Set interrupt base number

```

## Z-DOS INITIALIZATION SEQUENCE

---

```
MOV    AL,ICW3S3
OUT    ZM8259A+ICW3,AL ; Slave is connected to line 3
MOV    AL,ICW4UPM+ICW4SFN
OUT    ZM8259A+ICW4,AL ; Set processor to 8088, special fully nested, nonbuffered

MOV    AL,NOT (OCW1IM0 OR OCW1IM2 OR OCW1IM4 OR OCW1IM5 OR OCW1IM6)
OUT    ZM8259A+OCW1,AL ; Allow Fatal hardware, timer, serial A,
                       ; serial B, and keyboard/video interrupts
```

# ASCII CHART

---

OCT	DEC	HEX	CHAR	KEY	CTRL	DESCRIPTION
000	0	00	NUL	...	@	Null, tape feed.
001	1	01	SOH	...	A	Start of Heading.
002	2	02	STX	...	B	Start of text.
003	3	03	ETX	...	C	End of text.
004	4	04	EOT	...	D	End of transmission.
005	5	05	ENQ	...	E	Enquiry.
006	6	06	ACK	...	F	Acknowledge.
007	7	07	BEL	...	G	Rings Bell.
010	8	08	BS	BACK SPACE	H	Backspace; also FEB, Format Effector Backspace.
011	9	09	HT	TAB	I	Horizontal Tab.
012	10	0A	LF	LINE FEED	J	Line Feed: advances cursor to next line.
013	11	0B	VT	...	K	Vertical tab (VTAB).
014	12	0C	FF	...	L	Form feed to top of next page.
015	13	0D	CR	RETURN	M	Carriage Return to beginning of line.
016	14	0E	SO	...	N	Shift Out.
017	15	0F	SI	...	O	Shift In.
020	16	10	DLE	...	P	Data link escape.
021	17	11	DC1	...	Q	Device control 1: turns transmitter on (XON).
022	18	12	DC2	...	R	Device control 2.
023	19	13	DC3	...	S	Device control 3: turns transmitter off (XOFF).
024	20	14	DC4	...	T	Device control 4.
025	21	15	NAK	...	U	Negative acknowledge: also ERR (error).
026	22	16	SYN	...	V	Synchronous idle (SYNC).
027	23	17	ETB	...	W	End of transmission block.
030	24	18	CAN	...	X	Cancel (CANCL). Cancels current escape sequence.

## ASCII CHART

OCT	DEC	HEX	CHAR	KEY	CTRL	DESCRIPTION
031	25	19	EM	...	Y	End of medium.
032	26	1A	SUB	...	Z	Substitute.
033	27	1B	ESC	ESC	[	Escape.
034	28	1C	FS	...	\	File separator.
035	29	1D	GS	...	]	Group separator.
036	30	1E	RS	...	^	Record separator.
037	31	1F	US	...	-	Unit separator.
040	32	20	SP		...	Space (Spacebar).
041	33	21	!	!	...	Exclamation point.
042	34	22	"	"		Quotation mark.
043	35	23	#	#	...	Number sign.
044	36	24	\$	\$	...	Dollar sign.
045	37	25	%	%	...	Percent sign.
046	38	26	&	&	...	Ampersand.
047	39	27	'	'	...	Acute accent or apostrophe.
050	40	28	(	(	...	Open parenthesis.
051	41	29	)	)	...	Close parenthesis.
052	42	2A	*	*	...	Asterisk.
053	43	2B	+	+	...	Plus sign.
054	44	2C	,	,	...	Comma.
055	45	2D	-	-	...	Hyphen or minus sign.
056	46	2E	.	.	...	Period.
057	47	2F	/	/	...	Slash.
060	48	30	0	0	...	Number 0.
061	49	31	1	1	...	Number 1.
062	50	32	2	2	...	Number 2.
063	51	33	3	3	...	Number 3.
064	52	34	4	4	...	Number 4.
065	53	35	5	5	...	Number 5.
066	54	36	6	6	...	Number 6.
067	55	37	7	7	...	Number 7.
070	56	38	8	8	...	Number 8.
071	57	39	9	9	...	Number 9.
072	58	3A	:	:	...	Colon.
073	59	3B	;	;	...	Semicolon.
074	60	3C	<	<	...	Less than.



# ASCII CHART

---

OCT	DEC	HEX	CHAR	KEY	CTRL	DESCRIPTION	SYMBOL
075	61	3D	=	=	...	Equal sign.	
076	62	3E	>	>	...	Greater than.	
077	63	3F	?	?	...	Question mark.	
100	64	40	@	@	...	At sign.	
101	65	41	A	A	...	Letter A.	
102	66	42	B	B	...	Letter B.	
103	67	43	C	C	...	Letter C.	
104	68	44	D	D	...	Letter D.	
105	69	45	E	E	...	Letter E.	
106	70	46	F	F	...	Letter F.	
107	71	47	G	G	...	Letter G.	
110	72	48	H	H	...	Letter H.	
111	73	49	I	I	...	Letter I.	
112	74	4A	J	J	...	Letter J.	
113	75	4B	K	K	...	Letter K.	
114	76	4C	L	L	...	Letter L.	
115	77	4D	M	M	...	Letter M.	
116	78	4E	N	N	...	Letter N.	
117	79	4F	O	O	...	Letter O.	
120	80	50	P	P	...	Letter P.	
121	81	51	Q	Q	...	Letter Q.	
122	82	52	R	R	...	Letter R.	
123	83	53	S	S	...	Letter S.	
124	84	54	T	T	...	Letter T.	
125	85	55	U	U	...	Letter U.	
126	86	56	V	V	...	Letter V.	
127	87	57	W	W	...	Letter W.	
130	88	58	X	X	...	Letter X.	
131	89	59	Y	Y	...	Letter Y.	
132	90	5A	Z	Z	...	Letter Z.	
133	91	5B	[	[	...	Open brackets.	
134	92	5C	\	\	...	Reverse slash.	
135	93	5D	]	]	...	Close brackets.	
136	94	5E	^	^	...	Up arrow/caret.	<pre> {-----} {       } {   *   } {  * *  } { * * * } { * * * } { * * * } {   *   } {       } {-----} </pre>

# ASCII CHART

---

OCT	DEC	HEX	CHAR	KEY	CTRL	DESCRIPTION	SYMBOL
137	95	5F	_	_	...	Underscore.	<pre> (-----) (***** ) (***** ) (***** ) (***** ) (***** ) (***** ) (***** ) (***** ) (***** ) (***** ) (***** ) (-----)                 </pre>
140	96	60	`	`	...	Grave accent.	<pre> (-----) (  **  ) (  **  ) (  **  ) (  **  ) (  **  ) (  **  ) (  **  ) (  **  ) (  **  ) (  **  ) (  **  ) (-----)                 </pre>
141	97	61	a	a	...	Letter a.	<pre> (-----) (          ) (          ) (          ) (          ) (          ) (          ) (          ) (          ) (          ) (          ) (          ) (-----)                 </pre>
142	98	62	b	b	...	Letter b.	<pre> (-----) (  **  ) (  **  ) (  **  ) (  **  ) (  **  ) (  **  ) (  **  ) (  **  ) (  **  ) (  **  ) (  **  ) (-----)                 </pre>
143	99	63	c	c	...	Letter c.	<pre> (-----) (          ) (          ) (          ) (          ) (          ) (          ) (          ) (          ) (          ) (          ) (          ) (-----)                 </pre>
144	100	64	d	d	...	Letter d.	<pre> (-----) (  **  ) (  **  ) (  **  ) (  **  ) (  **  ) (  **  ) (  **  ) (  **  ) (  **  ) (  **  ) (  **  ) (-----)                 </pre>
145	101	65	e	e	...	Letter e.	<pre> (-----) (  **  ) (  **  ) (  **  ) (  **  ) (  **  ) (  **  ) (  **  ) (  **  ) (  **  ) (  **  ) (  **  ) (-----)                 </pre>

# ASCII CHART

---

OCT	DEC	HEX	CHAR	KEY	CTRL	DESCRIPTION	SYMBOL
146	102	66	f	f	...	Letter f.	<pre> (-----) (         ) (         ) (         ) (         ) (         ) (  fffff ) (  ff   ) (  ff   ) (  ff   ) (  ff   ) (  ff   ) (-----) </pre>
147	103	67	g	g	...	Letter g.	<pre> (-----) (         ) (         ) (         ) (         ) (         ) (  gffff ) (  gfff  ) (  gfff  ) (  gfff  ) (  gfff  ) (  gfff  ) (-----) </pre>
150	104	68	h	h	...	Letter h.	<pre> (-----) (         ) (         ) (         ) (         ) (         ) (  hffff ) (  hfff  ) (  hfff  ) (  hfff  ) (  hfff  ) (  hfff  ) (-----) </pre>
151	105	69	i	i	...	Letter i.	<pre> (-----) (  i i i i ) (  i i i i ) (  i i i i ) (  i i i i ) (  i i i i ) (  i i i i ) (  i i i i ) (  i i i i ) (  i i i i ) (  i i i i ) (  i i i i ) (-----) </pre>
152	106	6A	j	j	...	Letter j.	<pre> (-----) (  j j j j ) (  j j j j ) (  j j j j ) (  j j j j ) (  j j j j ) (  j j j j ) (  j j j j ) (  j j j j ) (  j j j j ) (  j j j j ) (  j j j j ) (-----) </pre>
153	107	6B	k	k	...	Letter k.	<pre> (-----) (         ) (         ) (         ) (         ) (         ) (  kffff ) (  kfff  ) (  kfff  ) (  kfff  ) (  kfff  ) (  kfff  ) (-----) </pre>
154	108	6C	l	l	...	Letter l.	<pre> (-----) (         ) (         ) (         ) (         ) (         ) (  lffff ) (  lfff  ) (  lfff  ) (  lfff  ) (  lfff  ) (  lfff  ) (-----) </pre>



# ASCII CHART

---

OCT	DEC	HEX	CHAR	KEY	CTRL	DESCRIPTION	SYMBOL
164	116	74	t	t	...	Letter t.	<pre> (-----) (  **  ) (  **  ) (  **  ) (  **  ) (***** ) (  **  ) (  **  ) (  **  ) (  **  ) (-----)                     </pre>
165	117	75	u	u	...	Letter u.	<pre> (-----) (  **  ) (  **  ) (  **  ) (  **  ) (***** ) (      ) (      ) (      ) (      ) (-----)                     </pre>
166	118	76	v	v	...	Letter v.	<pre> (-----) (  **  ) (  **  ) (  **  ) (  **  ) (***** ) (  **  ) (  **  ) (  **  ) (  **  ) (-----)                     </pre>
167	119	77	w	w	...	Letter w.	<pre> (-----) (  *  ** ) ( ** ** ) ( ** ** ) ( **** ) (  **  ) ( **** ) ( ** ** ) ( ** ** ) (  *  ** ) (-----)                     </pre>
170	120	78	x	x	...	Letter x.	<pre> (-----) (      x ) (     ** ) (    ** ) (   ** ) (  ** ) ( ** ) ( ** ) ( ** ) ( ** ) (-----)                     </pre>
171	121	79	y	y	...	Letter y.	<pre> (-----) (  **  ) (  **  ) (  **  ) (  **  ) (   ** ) (    ** ) (     ** ) (      x ) (-----)                     </pre>
172	122	7A	z	z	...	Letter z.	<pre> (-----) (***** ) (***** ) (      ) (      ) (      ) (      ) (      ) (      ) (-----)                     </pre>



## ESCAPE CODES

---

### Cursor Functions

ESC A	Cursor up
ESC B	Cursor down
ESC C	Cursor right
ESC D	Cursor left
ESC H	Cursor home
ESC I	Reverse index
ESC Y	Direct cursor addressing
ESC j	Save cursor position
ESC n	Cursor position report
ESC k	Set cursor to previously saved position

### Erasing and Editing

ESC E	Clear display and home cursor
ESC J	Erase to end of page
ESC K	Erase to end of line
ESC L	Insert line
ESC M	Delete line
ESC N	Delete character
ESC O	Exit insert character mode
ESC @	Enter insert character mode
ESC b	Erase to beginning of display
ESC I	Erase entire line
ESC o	Erase to beginning of line

# ESCAPE CODES

---

## Modes of Operation

ESC F	Enter graphics mode
ESC G	Exit graphics mode
ESC =	Enter alternate keypad mode
ESC >	Exit alternate keypad mode
ESC p	Enter reverse video mode
ESC q	Exit reverse video mode
ESC t	Enter keypad shifted mode
ESC u	Exit keypad shifted mode

ESC x Ps Set modes

Where Ps equals:

1	=	Enable 25th line
2	=	No key click
4	=	Block cursor
5	=	Cursor off
6	=	Keypad shifted
7	=	Enter alternate keypad mode
8	=	Auto line feed on receipt of CR
9	=	Auto CR of receipt of line feed
;	=	Nonblinking cursor
<	=	Disable keyboard auto repeat
?	=	Enable key expansion
@	=	Enable event driven (key up/down) mode



## ESCAPE CODES

---

ESC y Ps    Reset modes

Where Ps equals:

- 1    =    Disable 25th line
- 2    =    Enable key click
- 4    =    Underscore cursor
- 5    =    Cursor on
- 6    =    Keypad unshifted
- 7    =    Exit alternate keypad mode
- 8    =    No auto line feed
- 9    =    No auto CR
- ;    =    Blinking cursor
- <    =    Enable keyboard auto repeat
- ?    =    Disable key expansion
- @    =    Disable event driven  
          (key up/down) mode

ESC z        Reset to power-up configuration

### Additional Functions

ESC Z        Identify as VT52 (ESC/K)  
ESC #        Transmit page  
ESC ]        Transmit 25th line  
ESC ^        Transmit current line  
ESC \_        Transmit character at cursor  
ESC i 0      Zenith Identify Terminal Type  
ESC m fore back (ASCII digit to specify color)

Where fore and back equal:

- 0    =    Black
- 1    =    Blue
- 2    =    Red
- 3    =    Magenta
- 4    =    Green
- 5    =    Cyan
- 6    =    Yellow
- 7    =    White

## ESCAPE CODES

---

ESC {	Keyboard enable
ESC }	Keyboard disable
ESC v	Wrap-around at end of line
ESC w	Discard at end of line
ESC c	Key Click

The Computer will transmit the following sequences, but it will not respond to them if they are received by the Computer.

ESC J	Function Key F0
ESC S	Function Key F1
ESC T	Function Key F2
ESC U	Function Key F3
ESC V	Function Key F4
ESC W	Function Key F5
ESC P	Function Key F6
ESC Q	Function Key F7
ESC R	Function Key F8
ESC OI	Function Key F9
ESC OJ	Function Key F10
ESC OK	Function Key F11
ESC OL	Function Key F12

## ESCAPE CODES DEFINED

---

### Cursor Functions

#### **ESC A** Cursor Up

Moves the cursor up one line. If the cursor reaches the top line, it remains there, and no scrolling occurs.

#### **ESC B** Cursor Down

Moves the cursor down one line without changing columns. The cursor will not move past the bottom (24th) line and no scrolling will take place. Use Direct Cursor Addressing to move the cursor to line 25 — when line 25 is active.

#### **ESC C** Cursor Right

Moves the cursor one character position to the right. If the cursor is at the right end of the line, it will remain there.

#### **ESC D** Cursor Left

Moves the cursor one character position to the left (backspaces). If the cursor is at the start (left end) of a line, it will remain there.

#### **ESC H** Cursor Home

Moves the cursor to the first character position on the first line (home).

#### **ESC I** Reverse Index

Moves the cursor to the same horizontal position on the preceding line. If the cursor is on the top line, a scroll down is performed.

#### **ESC Y** Direct Cursor Addressing

Moves the cursor to a position on the screen by entering the escape code, the ASCII character which represents the line number, and the ASCII character which represents the column number.

## ESCAPE CODES DEFINED

---

The first line and the left column are both  $32_{10}$  (the smallest value of the printing characters) and increase from there. Since the lines are numbered from 1 to 25 (from top to bottom) and the columns from 1 to 80 (from left to right), you must add the proper line and column numbers to  $31_{10}$ . Then convert these decimal numbers to their equivalent ASCII characters and enter them in the following order:

ESC Y line # (ASCII character) column # (ASCII character)

If the line number entered is too high, the cursor will not move. If the column number is too high, the cursor will move to the end of the line.

This is the only way to move the cursor to the 25th line, but the 25th line must first be enabled.

### **ESC j** Save Cursor Position

The present cursor position is saved so the cursor can be returned here later when given the Set Cursor to Previously Saved Position command.

### **ESC n** Cursor Position Report

The Terminal reports the cursor position in the form of ESC Y line# column#.

### **ESC k** Set Cursor To Previously Saved Position

Returns the cursor to the position where it was when it received the Save Cursor Position command.

## ESCAPE CODES DEFINED

---

### Erasing And Editing

**ESC E** Clear Display And Home Cursor

Erases the entire screen, fills the screen with spaces, and places the cursor in the home position.

**ESC J** Erase To End Of Page

Erases all the information from the cursor (including the cursor position) to the end of the page.

**ESC K** Erase To End Of Line

Erases from the cursor (including the cursor position) to the end of the line.

**ESC L** Insert Line

Inserts a new blank line by moving the line that the cursor is on, and all following lines, down one line. Then the cursor is moved to the beginning of the new blank line.

**ESC M** Delete Line

Deletes the contents of the line that the cursor is on, places the cursor at the beginning of the line, moves all the following lines up one line, and adds a blank line at line 24.

**ESC N** Delete Character

Deletes the character at the cursor position and shifts any existing text that is to the right of the cursor one character position to the left.

**ESC O** Exit Insert Character Mode

Exits from the insert character mode.

## ESCAPE CODES DEFINED

---

**ESC @** Enter Insert Character Mode

Lets you insert characters or words into text already displayed on the screen. As you type in new characters, existing text to the right of the cursor shifts to the right. As each new character is inserted, the character at the end of the line is lost.

**ESC b** Erase To Beginning Of Display

Erases from the start of the screen to the cursor, and includes the cursor position.

**ESC I** Erase Entire Line

Erases all of the line, including the cursor position.

**ESC o** Erase To Beginning Of Line

Erases from the beginning of the line to the cursor, and includes the cursor position.

## ESCAPE CODES DEFINED

---

### Modes Of Operation

**ESC F** Enter Graphics Mode

Enters the graphics mode to display any of the 33 special symbols (26 lower-case keys and seven other keys) that correspond to the graphic symbols.

**ESC G** Exits Graphics Mode

Exits the graphics mode and returns to the display of normal characters.

**ESC =** Enter Alternate Keypad Mode

Enters the alternate keypad mode, which will then allow the keypad keys to transmit the following escape codes instead of the normal ones.

<u>KEY</u>	<u>ESCAPE CODE</u>
0	ESC ? p
1	ESC ? q
2	ESC ? r
3	ESC ? s
4	ESC ? t
5	ESC ? u
6	ESC ? v
7	ESC ? w
8	ESC ? x
9	ESC ? y
●	ESC ? n
ENTER	ESC ? M
—	ESC ? m

These special escape codes are user defined and must be recognized by your software.

## ESCAPE CODES DEFINED

---

**ESC >** Exit Alternate Keypad Mode

Exits the alternate keypad mode and returns to the transmission of normal character codes.

**ESC p** Enter Reverse Video Mode

Enters the reverse video mode so that characters are displayed as black characters on a white background.

**ESC q** Exit Reverse Video Mode

Exits the reverse video mode.

**ESC t** Enter Keypad Shifted Mode

Inverts the normal and shifted functions of the keypad. Now, if you hold down the SHIFT key, you will get a normally unshifted character.

**ESC u** Exit Keypad Shifted Mode

Exits the keypad shifted mode.



## ESCAPE CODES DEFINED

---

### Configuration

#### **ESC x Ps** Set Modes

Sets the following modes, where Ps equals:

- 1 = enable 25th line
- 2 = no key click
- 4 = block cursor
- 5 = cursor off
- 6 = keypad shifted
- 7 = enter alternate keypad mode
- 8 = auto line feed on receipt of CR
- 9 = auto CR on receipt of line feed
- ; = nonblinking cursor
- < = disable keyboard auto repeat
- ? = enable key expansion
- @ = enable event driven (key up/down) mode

#### **ESC y Ps** Reset Modes

Resets special modes, where Ps equals:

- 1 = disable 25th line
- 2 = enable key click
- 4 = underscore cursor
- 5 = cursor on
- 6 = keypad unshifted
- 7 = exit alternate keypad mode
- 8 = no auto line feed
- 9 = no auto CR
- ; = blinking cursor
- < = enable keyboard auto repeat
- ? = disable key expansion
- @ = disable event driven (key up/down) mode

# ESCAPE CODES DEFINED

---

## Additional Functions

### **ESC z** Reset To Power-Up Configuration

Nullifies all previously set escape modes and returns to the power-up configuration.

### **ESC Z** Identify As VT52 (ESC 1 K)

The Computer responds to the interrogation with ESC/K to indicate that it can perform as VT52.

### **ESC #** Transmit Page

Transmits lines 1 through 24. (The computer requires a special routine to use this feature.)

### **ESC ]** Transmit 25th Line

Transmits the 25th line. (The computer requires a special routine to use this feature.)

### **ESC ^** Transmit Current Line

Transmits the line that the cursor is currently located on. (The computer requires a special routine to use this feature.)

### **ESC \_** Transmit Character At Cursor

Transmits the character that the cursor is presently located at. (The computer requires a special routine to use this feature.)

### **ESC i 0** Zenith Identify Terminal Type

Interrogates the terminal for identification. A Z-100 family computer will respond with:

ESC i E Nn

Where Nn equals:

1 = one bank of VRAM

3 = three banks of VRAM

A = 32 k byte VRAM parts

B = 64 k byte VRAM parts

## ESCAPE CODES DEFINED

---

### **ESC m** Fore Back

Specifies colors for foreground and background of display, where fore and back equal:

- 0 = black
- 1 = blue
- 2 = red
- 3 = magenta
- 4 = green
- 5 = cyan
- 6 = yellow
- 7 = white

### **ESC {** Keyboard Enable

Enables the keyboard after it was inhibited by an Keyboard Disabled command.

### **ESC }** Keyboard Disable

Inhibits the output of the keyboard.

### **ESC v** Wrap-Around At End Of Line

The 81st character on a line is automatically placed in the first character position on the next line. The page scrolls up if necessary.

### **ESC w** Discard At End Of Line

After the 80th character in a line, the characters overprint. Therefore, only the last character received will be displayed in position 80.

## ESCAPE CODES DEFINED

---

**ESC J** Function Key F0

Transmits a unique escape code to perform a user-defined function. The computer will not respond to this code if it is received.

**ESC S** Function Key F1

Same as above.

**ESC T** Function Key F2

Same as above.

**ESC U** Function Key F3

Same as above.

**ESC V** Function Key F4

Same as above.

**ESC W** Function Key F5

Same as above.

**ESC P** Function Key F6

Same as above.

**ESC Q** Function Key F7

Same as above.

**ESC R** Function Key F8

Same as above.

**ESC OI** Function Key F9

Same as above.

**ESC OJ** Function Key F10

Same as above.

**ESC OK** Function Key F11

Same as above.

**ESC OL** Function Key F12

Same as above.

## KEY CODE CHART

---

After a key is detected as being down, the keyboard encoder places a byte on its data bus which represents only the depressed key. The codes for some of the keys depend on the state of the "modifier" keys — SHIFT (right or left), CTRL (control), and CAPS LOCK. Some keys are not affected by any of the modifiers, such as the DELETE key. Its code (7F) is always the same, such as the DELETE key. It's code (7F) is always the same, regardless of the modifier key's positions. Other keys are affected by all of the modifiers, such as the "A" key.

In the following table, an "NC" under a modifier indicates that no code is generated for that key.

The CAPS LOCK column has a Y (yes) or N (no) to indicate if the CAPS LOCK key affects the output code or not. The CAPS LOCK key functions as a SHIFT key, but only for the alphabet keys.

Each key has a code for when it is pushed down. However, in its event-driven mode (key up/down mode), each key also has a different code for when it starts back up again. These are listed as Down Codes and Up Codes. (The "up code" equals the "down code" plus 80 hex.)

Key	Not Shifted	Shifted	Control	Control Shift	Caps Lock (Yes/No)	Down Code	Up Code
) 0	30	29	30	29	N	5B	DB
! 1	31	21	31	21	N	57	D7
@ 2	32	40	32	00	N	56	D6
# 3	33	23	33	23	N	55	D5
\$ 4	34	24	34	24	N	54	D4

## KEY CODE CHART

Key	Not Shifted	Shifted	Control	Control Shift	Caps Lock (Yes/No)	Down Code	Up Code
% 5	35	25	35	25	N	53	D3
^ 6	36	5E	36	1E	N	52	D2
& 7	37	26	37	26	N	51	D1
* 8	38	2A	38	2A	N	50	D0
( 9	39	28	39	28	N	5A	DA
A	61	41	01	01	Y	07	87
B	62	42	02	02	Y	13	93
C	63	43	03	03	Y	15	95
D	64	44	04	04	Y	05	85
E	65	45	05	05	Y	0D	8D
F	66	46	06	06	Y	04	84
G	67	47	07	07	Y	03	83
H	68	48	08	08	Y	02	82
I	69	49	09	09	Y	08	88
J	6A	4A	0A	0A	Y	01	81
K	6B	4B	0B	0B	Y	00	80
L	6C	4C	0C	0C	Y	10	90

## KEY CODE CHART

Key	Not Shifted	Shifted	Control	Control Shift	Caps Lock (Yes/No)	Down Code	Up Code
M	6D	4D	0D	0D	Y	11	91
N	6E	4E	0E	0E	Y	12	92
O	6F	4F	0F	0F	Y	19	99
P	70	50	10	10	Y	1A	9A
Q	71	51	11	11	Y	0F	8F
R	72	52	12	12	Y	0C	8C
S	73	53	13	13	Y	06	86
T	74	54	14	14	Y	0B	8B
U	75	55	15	15	Y	09	89
V	76	56	16	16	Y	14	94
W	77	57	17	17	Y	0E	8E
X	78	58	18	18	Y	16	96
Y	79	59	19	19	Y	0A	8A
Z	7A	5A	1A	1A	Y	17	97
BACKSPACE	08	08	08	08	N	5F	DF
TAB	09	09	09	09	N	4E	CE
LINE FEED	0A	0A	0A	0A	N	44	C4
RETURN	0D	0D	0D	0D	N	4C	CC

## KEY CODE CHART

Key	Not Shifted	Shifted	Control	Control Shift	Caps Lock (Yes/No)	Down Code	Up Code
ESC	1B	1B	1B	1B	N	4F	CF
SPACE	20	20	20	20	N	45	C5
" '	27	22	27	22	N	48	C8
< ,	2C	3C	2C	3C	N	4D	CD
— _	2D	5F	2D	1F	N	5C	DC
> .	2E	3E	2E	3E	N	4A	CA
? /	2F	3F	2F	3F	N	4B	CB
: ;	3B	3A	3B	3A	N	49	C9
+ =	3D	2B	3D	2B	N	5D	DD
— [	5B	7B	1B	7B	N	59	D9
 \ /	5C	7C	1C	7C	N	43	C3
} ]	5D	7D	1D	7D	N	58	D8
~ `	60	7E	60	7E	N	5E	DE



## KEY CODE CHART

Key	Not Shifted	Shifted	Control	Control Shift	Caps Lock (Yes/No)	Down Code	Up Code
DELETE	7F	7F	7F	7F	N	42	C2
ENTER	8D	CD	8D	CD	N	38	B8
HELP	95	D5	95	C5	N	46	C6
F0	96	D6	96	D6	N	27	A7
F1	97	D7	97	D7	N	26	A6
F2	98	D8	98	D8	N	25	A5
F3	99	D9	99	D9	N	24	A4
F4	9A	DA	9A	DA	N	23	A3
F5	9B	DB	9B	DB	N	22	A2
F6	9C	DC	9C	DC	N	21	A1
F7	9D	DD	9D	DD	N	20	A0
F8	9E	DE	9E	DE	N	29	A9
F9	9F	DF	9F	DF	N	2A	AA
F10	A0	E0	A0	E0	N	2B	AB
F11	A1	E1	A1	E1	N	2C	AC
F12	A2	E2	A2	E2	N	2D	AD

## KEY CODE CHART

Key	Not Shifted	Shifted	Control	Control Shift	Caps Lock (Yes/No)	Down Code	Up Code
D CHR I CHR	A3	E3	A3	E3	N	2E	AE
D LINE I LINE	A4	E4	A4	E4	N	2F	AF
(up arrow)	A5	E5	A5	E5	N	3B	BB
(down arrow)	A6	E6	A6	E6	N	3A	BA
(right arrow)	A7	E7	A7	E7	N	33	B3
(left arrow)	A8	E8	A8	E8	N	3F	BF
HOME	A9	E9	A9	E9	N	37	B7
BREAK	AA	EA	AA	EA	N	47	C7
- (keypad)	AD	ED	AD	ED	N	39	B9
(keypad)	AE	EE	AE	EE	N	40	C0
0 (keypad)	B0	F0	B0	F0	N	41	C1
1 (keypad)	B1	F1	B1	F1	N	34	B4
2 (keypad)	B2	F2	B2	F2	N	3C	BC
3 (keypad)	B3	F3	B3	F3	N	30	B0
4 (keypad)	B4	F4	B4	F4	N	35	B5
5 (keypad)	B5	F5	B5	F5	N	3D	BD
6 (keypad)	B6	F6	B6	F6	N	31	B1

## KEY CODE CHART

---

Key	Not Shifted	Shifted	Control	Control Shift	Caps Lock (Yes/No)	Down Code	Up Code
7 (keypad)	B7	F7	B7	F7	N	36	B6
8 (keypad)	B8	F8	B8	F8	N	3E	BE
9 (keypad)	B9	F9	B9	F9	N	32	B2
FAST REPEAT	NC	NC	NC	NC	N	60	E0
CAPS LOCK	NC	NC	NC	NC	N	61	E1
SHIFT (right)	NC	NC	NC	NC	N	62	E2
CTRL	NC	NC	NC	NC	N	63	E3
SHIFT (left)	NC	NC	NC	NC	N	64	E4
RESET	NC	NC	(NC) Resets Computer	(NC) Resets Computer	N	NC	NC

## KEYPAD CODE CHART

## Keypad Codes (key expansion enabled)

Key(s) Pressed:	MODES			
	Normal Unshifted	Normal Shifted	Alternate Unshifted	Alternate Shifted
ENTER	ENTER	ENTER	ESC?M	ENTER
.	.	.	ESC?n	.
-	-	-	ESC?m	-
0	0	0	ESC?p	0
1	1	ESCL	ESC?q	ESCL
2	2	ESCB	ESC?r	ESCB
3	3	ESCM	ESC?s	ESCM
4	4	ESCD	ESC?t	ESCD
5	5	ESCH	ESC?u	ESCH
6	6	ESCC	ESC?v	ESCC
7	7	* ESC@/ESCO	ESC?w	* ESC@/ESCO
8	8	ESCA	ESC?x	ESCA
9	9	ESCN	ESC?y	ESCN
SHIFT ENTER	ENTER	ENTER	ENTER	ESC?M
SHIFT .	.	.	.	ESC?n
SHIFT -	-	-	-	ESC?m
SHIFT 0	0	0	0	ESC?p
SHIFT 1	ESCL	1	ESCL	ESC?q
SHIFT 2	ESCB	2	ESCB	ESC?r
SHIFT 3	ESCM	3	ESCM	ESC?s
SHIFT 4	ESCD	4	ESCD	ESC?t
SHIFT 5	ESCH	5	ESCH	ESC?u
SHIFT 6	ESCC	6	ESCC	ESC?v
SHIFT 7	* ESC@/ESCO	7	* ESC@/ESCO	ESC?w
SHIFT 8	ESCA	8	ESCA	ESC?x
SHIFT 9	ESCN	9	ESCN	ESC?y

\* Toggles between codes

# FUNCTION KEY CODE CHART

---

## Function Key Codes (key expansion enabled)

Key	Unshifted	Shifted
F0	ESC J	ESC E
F1	ESC S	ESC 1 A
F2	ESC T	ESC 1 B
F3	ESC U	ESC 1 C
F4	ESC V	ESC 1 D
F5	ESC W	ESC 1 E
F6	ESC P	ESC 1 F
F7	ESC Q	ESC 1 G
F8	ESC R	ESC 1 H
F9	ESC O I	ESC 1 I
F10	ESC O J	ESC 1 J
F11	ESC O K	ESC 1 K
F12	ESC O L	ESC 1 L
I CHR	*ESC @ / ESC 0	—
D CHR	—	ESC N
DEL LINE	—	ESC M
INS LINE	ESC L	—
↑	ESC A	ESC A
↓	ESC B	ESC B
→	ESC C	ESC C
←	ESC D	ESC D
HOME	ESC H	ESC H
BREAK	ESC	ESC
HELP	ESC ~	ESC ~

\* Toggles between codes

---

# Index

## A

Address/Data circuits, 2.65  
  Address latches, 2.66  
  Data latches, 2.67  
  Extended addressing, 2.69  
  General, 2.65

Address Multiplexer, 2.50

ASCII chart, 10.30

## B

BDOA, 4.55

Block Diagrams,

  CPU block diagram, 2.19

  Interrupt block diagram, 2.21

  I/O block diagram, 2.22

  Keyboard block diagram, 2.22

  Memory block diagram, 2.20

Black Level Control, 4.4

Brite control, 5.5

## C

Cabinet, 9.1

Cabinet top, 1.5

Cables, 9.1

Cascade amplifier, 5.3

Chassis, 9.1

Chassis, cabinet, & cables, 9.1

  Cable location/description, 9.12

  Circuit boards, & hardware, 9.17

  Replacement Parts List, 9.2

    All-In-One, 9.2

    Low profile, 9.8

Circuit descriptions,

  Main board, 2.23

  Video deflection board, 5.2

  Video logic board, 4.48

Color display, 4.6

Color output, 4.55

Composite, 4.56

Configuration, 10.48

Contrast control, 4.4

CPU read, 4.62

CPU write, 4.62

CRT-C, 4.8

CRT-C read, 4.63

CRT-C registered, 4.40

Cursor functions, 10.42

## D

Differential amplifier, 5.2

DIP switch port (FF), 2.8

DIP switch S-101, 2.3

DIP switch select circuits, 2.40

DS1 floppy disk controller board, 6.5

DS1 floppy disk controller board, 6.8

# INDEX

---

- Disassembly, 1.5
  - Display/Front panel assembly, 1.5
  - Keyboard, 1.6
  - Keyboard shell, 1.6
  - Power supply, 1.7
  - S-100 card cage, 1.7
  - Video logic circuit board, 1.8
  - Main board, 1.9
- Disk controller board, 1.4
- Drive interface connectors, 6.13
- Dynamic memory, 2.48
  - Address multiplexer, 2.50
  - Dynamic RAM, 2.49
  - General, 2.48
  - Memory circuit waveforms, 2.53
  - Memory map decoder, 2.51

## E

- E-clock, 2.81
- Editing, 10.45
- Encoder output codes, 3.12
- Erasing, 10.45
- Escape codes, 10.38
- Escape codes defined, 10.42
- Event-driven mode, 3.2
- Exploded view — All-In-One, 9.3
- Exploded view — Low profile, 9.9
- Extended addressing, 2.68

## F

- Flash, 4.24, 4.58
- Floppy disk controller card, 1.5

- Floppy disk controller, 6.1
  - Address lines, 6.24
  - Assembly language code, 6.16
  - Bus interface, 6.21
  - Calibration, 6.35
  - Card clock speed, 6.3
  - Circuit description, 6.23
  - Control latch, 6.22
  - Control latch bit definitions, 6.9
  - Control lines, 6.24
  - Controller, 6.21
  - Data in, 6.23
  - Data out, 6.23
  - Data precompensation, 6.30
  - Data read, 6.28
  - Data separation, 6.21, 6.30
  - Data separator calibration, 6.37
  - Data write, 6.28
  - Description, 6.2
  - Drive interface, 6.21, 6.30
  - Drive interface connectors, 6.13
  - DS1, 6.5, 6.8
  - Interconnect pins, 6.50
  - Interleaving, 6.12
  - I/O port assignments, 6.7
  - Port bit definitions, 6.8
  - Port address selection, 6.5
  - Power up, 6.24
  - Precompensation, 6.21
  - Precompensation options, 6.11
  - RDY delay, 6.29
  - Read status latch, 6.26
  - Read status register (1797), 6.27
  - Ready line, 6.24
  - Replacement Parts List, 6.38
  - S-100 bus connector, 6.15

S-100 bus interface, 6.23  
 Semiconductor identification, 6.39  
 Signal definitions, 6.50  
 Status port, 6.21  
 Status port bit definitions, 6.10  
 Theory of operation, 6.21  
 Track formats, 6.11  
 Troubleshooting, 6.32  
 Vector interrupt lines, 6.4, 6.24  
 Write command register (1797), 6.28  
 Write control latch, 6.27  
 X-Ray View, 6.49

Floppy drives 5 1/4", 7.1  
 Cable connections, 7.5  
 Description, 7.2  
 Diskette handling, 7.6  
 Diskette loading, 7.6  
 Operation, 7.6  
 Programming, 7.3  
 Programming plugs, 7.4  
 Terminator IC's, 7.4  
 Write-protect, 7.7  
 Flyback transformer, 5.2

## G

GDOTA, 4.55

## H

Handshake, 2.84, 2.87  
 High address latch (FD), 2.10  
 Horizontal sync, 5.2

## I

Initialization sequence, 10.14  
 Interconnect pin definitions,  
   Main board, 2.137  
     Keyboard, 2.139  
     Light pen, 2.139  
     Parallel port, 2.138  
     RS-232, 2.137  
     S-100 bus, 2.137  
 Video logic board, 2.141  
 Video logic board, 4.109  
 Power Supply connectors, 2.144  
 Interleaving, 6.12  
 Interrupt 8259A (F0-F3), 2.15  
 Interrupt circuitry, 2.69  
   General, 2.69  
   Interrupt routine, 2.74  
   Maskable interrupt sequence, 2.70  
   Nonmaskable interrupt sequence, 2.73  
 I/O circuitry, 2.82  
   I/O port decoder, 2.90  
   General, 2.82  
   Parallel port, 2.86  
   Printer port, 2.87  
   Light pen port, 2.88  
   Serial port A, 2.82  
   Serial port B, 2.85  
   Video interrupt port, 2.89

IO.SYS, 10.18

## J

Jumpers — floppy disk controller, 6.3, 6.4  
 Jumpers — main board, 2.4  
 Jumpers — video logic board, 4.3



# INDEX

---

## K

- Keyboard, 2.75
  - Encoder, 3.1
  - Encoder output codes, 3.12
  - Functions, 3.2
  - General, 2.75
  - Layout, 3.19
  - Matrix, 3.11
  - 8041 pin-out, 2.75
  - Theory of Operation, 3.8
  - Troubleshooting, 3.10
- Key code chart, 10.52

## L

- Light pen, 4.7, 10.13
- Light pen port, 2.17, 2.88
- Loader, 10.13

## M

- Main board, 1.4
  - Address/Data circuits, 2.65
  - Block diagram — CPU, 2.19
  - Block diagram — interrupt, 2.21
  - Block diagram — I/O, 2.22
  - Block diagram — keyboard, 2.22
  - Block diagram — memory, 2.20
  - Circuit Description, 2.23
  - Description, 2.2
  - DIP switch S101, 2.3
  - DIP switch select circuits, 2.40
  - E-clock, 2.78
  - I/O circuitry, 2.23, 2.83
  - Interconnect pin definitions, 2.137
  - Interrupt circuitry, 2.69
  - Interrupts 8259A, 2.21

- Jumpers, 2.4
- Keyboard, 2.75
- Keyboard encoder, 2.23, 3.1
  - Auto repeat, 3.3
  - Command summary, 3.6
  - Event-driven mode, 3.3
  - FIFO, 3.3
  - I/O protocol, 3.4
  - Key click, 3.3
  - Power configuration, 3.3
  - Programming specification, 3.4
- Map selection, 2.61
- Memory, 2.20
- Memory control latch, 2.48
- Memory control latch port (FC), 2.11
- Memory map, 4.59
- Microprocessor 8085, 2.23
  - General, 2.23
  - Pin-out description, 2.23
  - Timing, 2.26
- Microprocessor 8088, 2.28
  - General, 2.28
  - Pin-out description, 2.29
  - Timing, 2.32
- Microprocessor status code, 2.43
- Parity circuits, 2.59
- Processor swap port, 2.33
  - Auto swap, 2.37
  - General, 2.33
  - Swap interrupt, 2.38
  - Swap timing, 2.34
- Refresh circuits, 2.55
- Reset circuits, 2.39
  - Keyboard reset, 2.40
  - Power-up reset, 2.39
- Replacement Parts List, 2.92
- Semiconductor Identification, 295
- S-100 bus control output circuits, 2.45
- S-100 bus status, 2.41
- System monitor ROM, 2.63
- Theory of Operation, 2.18

Timer, 2.78  
 Wait timing, 2.44  
 X-Ray Views, 2.137  
 Master 8259A, 2.15  
 Memory circuit waveforms, 2.53  
 Memory mapping, 10.6  
 Modes of operation, 10.46  
 Monochrome, 4.7, 4.22  
 Monochrome output, 4.55

## P

Parallel port, 10.13  
 Parallel printer port, 2.17  
 Parity, 2.12, 10.7  
 Parity circuits — main board, 2.59  
 Phantom line, 2.52, 2.64, 4.23  
 Pixel, 4.5, 4.40  
 Polling, 3.5  
 Port addresses, 2.6, 10.11  
 Port D8, 4.32  
 Power supply, 5.3  
 Power supply, 8.1  
   Specifications, 8.3  
 Processor swap port (FE), 2.9  
 Programming data, 10.1  
   General information, 10.3  
     8085, 10.3  
     8088, 10.3  
     Floppy disk controller, 6.7  
     Interrupt structure, 10.4  
     Memory mapping, 10.6  
     Parity, 10.7  
     Processor swapping, 10.5  
     RAM, 10.3  
     ROM, 10.3  
     Timer, 10.8

Programming Information, 2.6  
   DIP switch port (FF), 2.8  
   High address latch (FD), 2.10  
   Interrupt 8259A (F0-F3), 2.15  
   Light pen port, 2.17  
   Memory control latch port (FC), 2.11  
   Parallel port 68A21 (E0-E3), 2.16  
   Port addresses, 2.6  
   Port bit definitions, 2.7  
   Printer parallel port, 2.17  
   Processor swap port (FE), 2.9  
   Timer 8253 bit definitions, 2.14  
   Timer 8253 status port (FB), 2.13

## R

RDOTA, 4.55  
 Recalibration — video deflection board, 5.5  
 Refresh circuits, — main board, 2.55  
 Refresh clock, 2.55  
 Relative memory locations, 4.51  
 Replacement Parts List,  
   Floppy disk controller board, 6.38  
   Main board, 2.92  
   Video deflection board, 5.8  
   Video logic board, 4.70

## S

Semiconductor Identification  
   Floppy disk controller board, 6.39  
   Main board, 2.95  
   Video logic board, 4.72  
  
 S-100 bus status circuits, 2.41  
 S-100 control output circuits, 2.45

# INDEX

---

- Slave 8259A, 2.15
  - Specifications,
    - Power supply, 8.3
  - System monitor ROM, 2.63
    - Addressing, 2.63
    - Phantom line, 2.64
- T**
- Theory of Operation — main board, 2.18
  - Timer, 2.78
  - Timer status port (FB), 2.13
  - Troubleshooting,
    - Floppy disk controller board, 6.32
    - Video logic board, 4.69
    - Video deflection board, 5.4
- U**
- UPI — 8041A, 2.75
- V**
- Vertical control register, 4.30
  - Vertical size, 5.6
  - Video deflection board, 5.1
    - Circuit Description, 5.2
      - High voltage power supply, 5.3
      - Horizontal circuits, 5.2
      - Power supply, 5.3
      - Vertical circuits, 5.2
      - Video amplifier, 5.3
  - Recalibration, 5.5
  - Replacement parts list, 5.8
  - Troubleshooting, 5.4
  - X-Ray Views, 5.11
- Video logic board, 1.4, 4.1
    - Black level control, 4.4
    - Circuit Description, 4.48
      - CPU-video communications, 4.57
      - Light pen circuits, 4.68
      - Relative memory locations, 4.51
      - Timing, 4.65
      - Video arbitration, 4.66
      - Video output, 4.55
      - Video processing circuits, 4.48
  - Replacement Parts List, 4.70
  - Semiconductor identification, 4.72
  - Troubleshooting, 4.69
  - X-Ray Views, 4.109
  - Color display, 4.6
  - Contrast control, 4.4
  - Conversion from character based
    - to pixel based display, 4.11
  - CRT-C, 4.12
  - Description, 4.2
    - Interconnect pin definitions, 4.109
    - Jumpers, 4.3
    - Light pen, 4.7
    - Light pen, 4.24
      - Software considerations, 4.26
    - Matrix scheme, 4.5
    - Programming data, 4.29
      - Clearing the screen, 4.45
      - CRT-C register, 4.39
      - Port addresses, 4.29
      - Video control register, 4.30
  - Read data buffers, 4.61
  - Theory of Operation, 4.5
    - Video RAM mapping module, 4.15
    - Video RAM, 4.22
    - Video system, 4.8

**W**

Wait state, 2.44  
Wait control, 5.6  
Width control, 2.44

**X**

X-Ray views,  
    Floppy disk controller board, 6.49  
    Main board, 2.136  
    Video deflection board, 5.11  
    Video logic board, 4.106

**Z**

Z-DOS drive mapping, 10.21  
Z-DOS initialization sequence, 10.14

