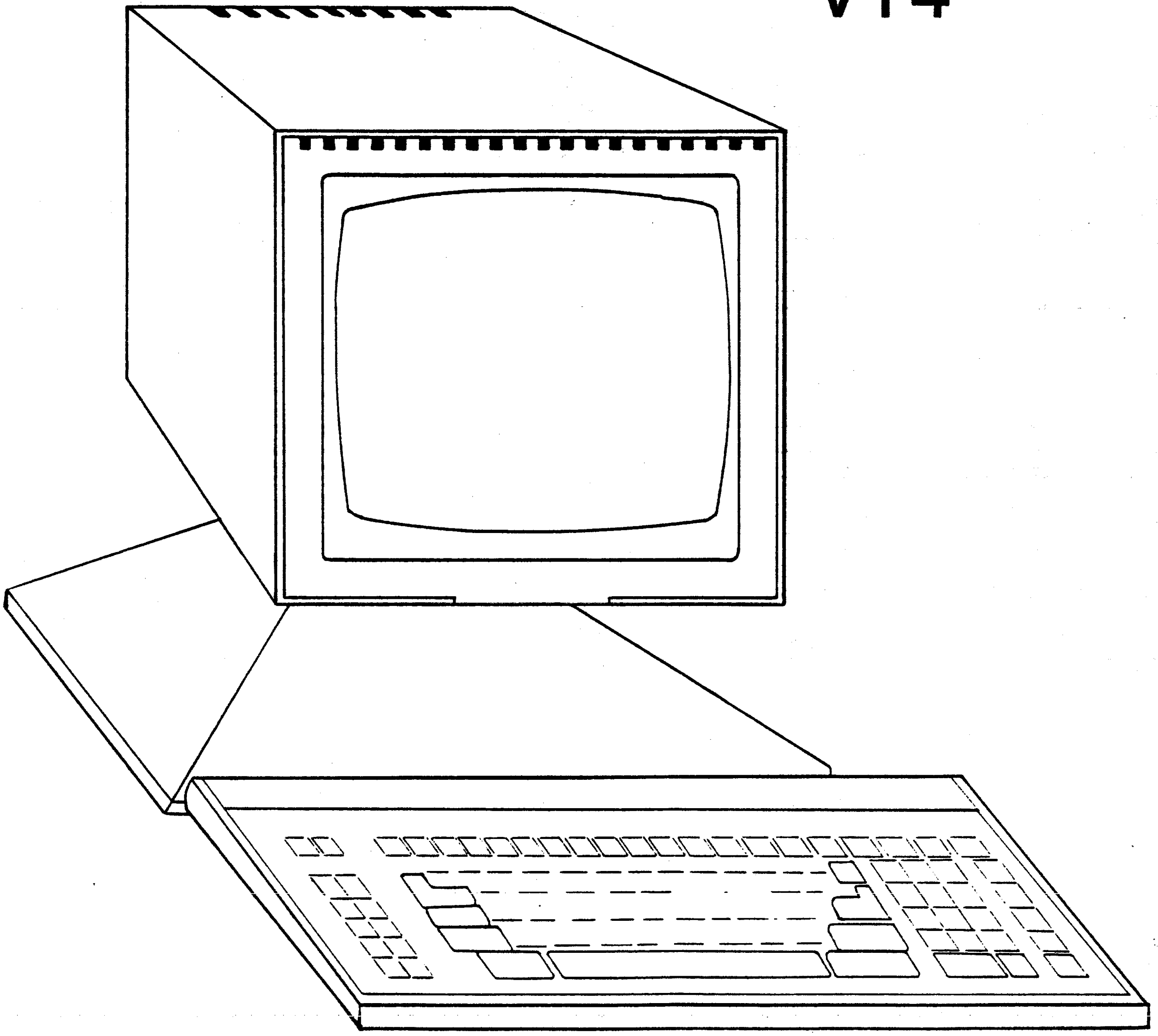
MOHAWK DATA SCIENCES
SERVICE DIVISION

THEORY OF OPERATION

VT4



M-4948

(TD-4103)

TECHNICAL MANUAL

THEORY OF OPERATION
VT4

FIRST EDITION

Notice: This document is supplemented by the document called SCHEMATICS for THEORY OF OPERATION VT4 (TD-4103s).

THEORY OF OPERATION - VT4

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

EQUIPMENT CHARACTERISTICS

1.0 GENERAL

The VT4 is a terminal that is able to communicate with any Qantel computer in the same manner as the VT3. In addition, the VT4 has 64K of RAM which enables the terminal to act as a microcomputer. At this writing, the primary function of the VT4's RAM will be to store instructions and data to run *QICWORD, which is Qantel's sophisticated word processing program.

*QICWORD as well as some other proposed programs will be downloaded to the VT4. Users of these programs can then be working while other users are concurrently using BEST.

1.1 ELECTRICAL REQUIREMENTS

A.C.

Voltages and Tolerances

2A - 254 W Max.	1A - 240 W Max.
10A surge on powerup:	5A surge on powerup:
100vac ± 10vac	200vac ± 20vac
115vac ± 11.5vac*	220vac ± 22vac
127vac ± 12.7vac	230vac ± 23vac
240vac ± 24vac	
* Normal U.S.A. installation	

Frequency and Tolerance

60 Hz ± 0.5 Hz
50 Hz ± 0.5 Hz

D.C.

+12v ± 0.6v, 2.0A
-12v ± 0.6v, 0.1A
+5v ± 0.25v, 3.25A

1.2 ENVIRONMENTAL REQUIREMENTS

Ambient Room Temperature - Operating: 50F - 95F (10C - 35C)

Shipping and Storage Temperature: -40F - 150F (-40C - 66C)

Relative Humidity - Operating: 10% - 90%
with no condensation.

Relative Humidity - Shipping & Storage: 0% - 95%
with no condensation.

Altitude - Operating: 0 - 7875 ft. (0 - 2400m)

Altitude - Shipping & Storage: 0 - 29500 ft. (0 - 9000m)

1.3 PHYSICAL DIMENSIONS

The VT4R with the keyboard, requires a minimum desk space of 20 inches (51.5cm) in width by 25 inches (67cm) in depth. A vertical clearance of at least 18 inches (46.4cm) should be allowed so the terminal can be tilted without interference.

2.0 GENERAL

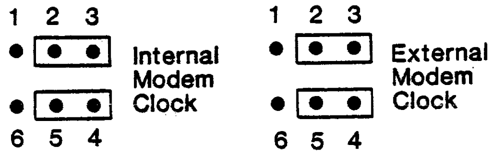
The VT4 connects to the IOU39Q in the same manner as the VT3, as far as cables and terminators are concerned. JMP1 and JMP2 on the VT4 logic board must be installed when line drivers and/or non-powered hub units are used, so that these devices will have power. JMP1 and JMP2 should be installed for all USA installations.

2.1 JUMPER SETTINGS (Figure 2-1)

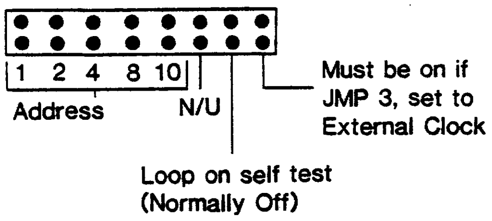
Jumpers on the VT4 logic board should be set as described below:

JMP 1, 2 +12v and -12v respectively on the RS-232 line. These jumpers should be in place for United States installations.

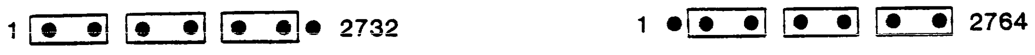
JMP 3. This allows selection of the internal or an external QSP clock. The usual setting allows for the internal modem clock.



JMP 4. This is the QSP "A" Port Address Jumper. The first five positions, as they are counted from the outside edge of the board toward the inside of the board, represent the least to the most significant bits respectively.

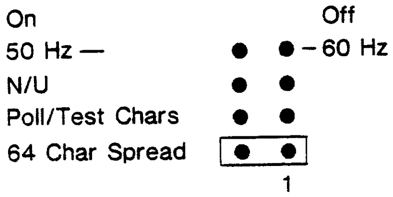

1 2 4 8 10
Address
N/U
Must be on if JMP 3, set to External Clock
Loop on self test (Normally Off)

JMP 5. This is the ROM Select Jumper. They should all be installed for the ROMs currently being used in the VT4.

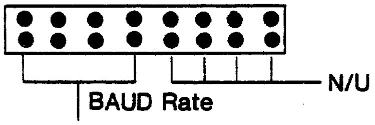


JMP 6. This is the Device B (AUX Port 1) Address Jumper. Bits 1 through 5 represent the QSP address of the device on the AUX 1 Port. Bits 6,7, and 8 represent the baud rate for the device on the AUX 1 Port. (software does not support JMP 6 at this writing).

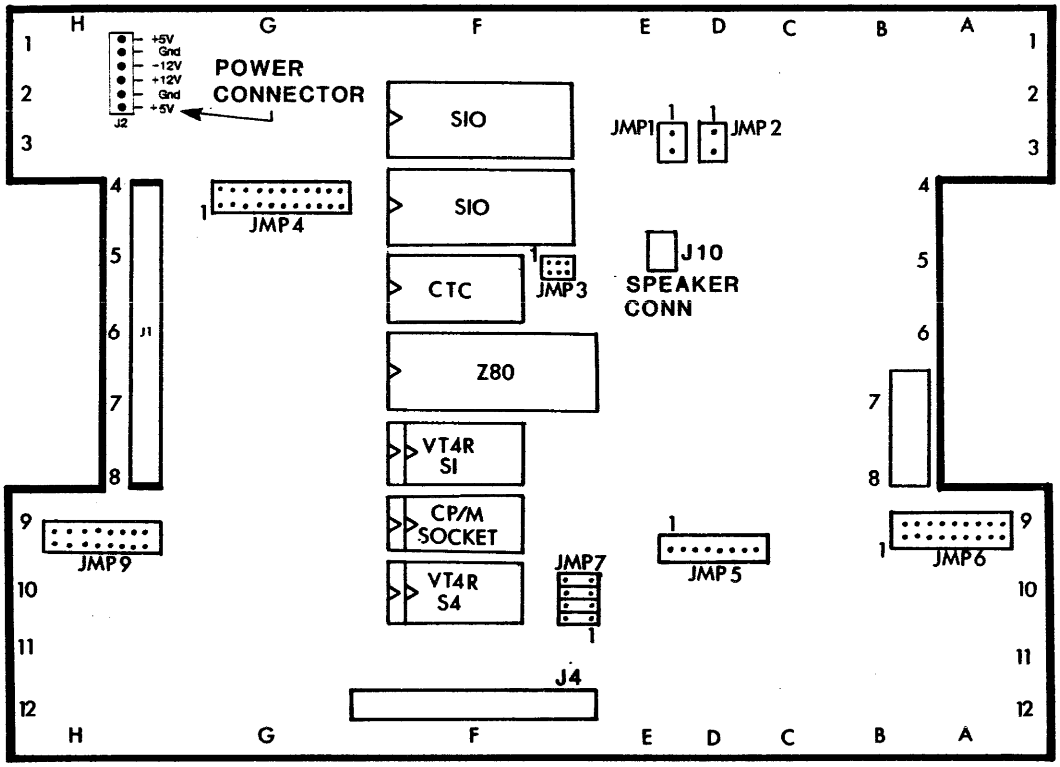
JMP 7. This is a combination test and a 50Hz/60Hz select jumper. The top pair of pins selects 50Hz when installed. The second pair of pins is not used. The third pair of pins enables the poll character to appear in the bottom corner of the screen (and disables Loop on Self-Test on JMP 4). Finally, the fourth pair of pins enables the character display to be spread properly across the full width of the screen when the terminal is in 64 character mode.



JMP 9. This selects the QSP baud rate for the VT4. Only bits 5,6,7, and 8 are used (bit 8 is the closest to the edge of the board).



0000 =	19.2K	1 = Jumper
0001 =	9.6K	0 = No Jumper
0011 =	4.8K	
0101 =	2.4K	
1000 =	1.2K	
1001 =	600	
1010 =	300	



CPU BOARD

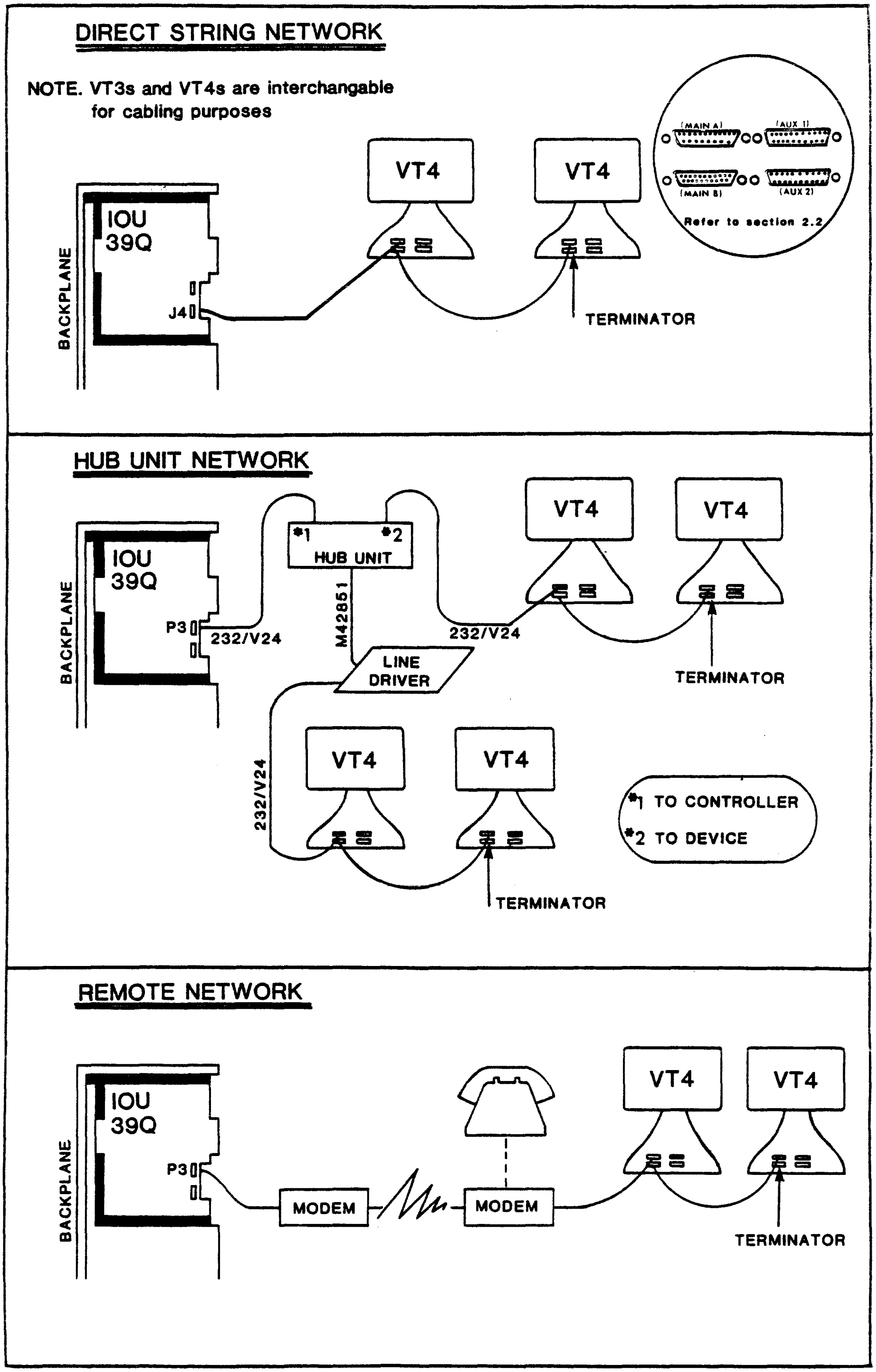
Figure 2-1

2.2 CABLING

When viewing the VT4 from the rear, the upper left RS-232 connector ("Main A" in the inset in Figure 2-2) is for the incoming cable from the IOU 39Q or for the incoming cable from the IOU 39Q string. The lower left connector (Main B) is the outgoing cable to the next terminal or RDI box on the string. If the VT4 is the last device on the string, a terminator must be installed in the lower left RS-232 connector. The other two RS-232 connectors (Aux 1 and Aux 2) are not used by the software at this time.

NOTE THAT THE CPU LOGIC BOARD CANNOT BE DISCONNECTED FROM THE VT4 WITHOUT BREAKING THE QSP STRING.

Cabling functions are the same as for the VT3.



NOTE. VT3s and VT4s are interchangable for cabling purposes

Figure 2-2

3.0 POWER SUPPLY ADJUSTMENTS

Voltage test points on the CPU logic board can be reached from the rear of the terminal after the top cover is removed. The test points will be at the lower left corner of the board as viewed from the rear of the terminal.

Since the board is installed upside-down, the CPU voltage test points (which are behind power connector J2) are inverted with respect to the way they are shown in Figure 2-1.

Power on the VT4 and measure the +5v and the +12v and -12v to ground using the voltage test points on the back of the CPU board. The voltages should be: +5v±0.25v and +12v±0.6v. If the +5v or +12v are not within tolerance, the power supply should be adjusted (there is no -12v adjustment).

The power supply is attached to the bottom pan of the VT4. The power supply adjustment procedure is as follows:

1. Remove the top cover of the VT4.
2. Lay the terminal carefully down on its side.
3. Remove the seven screws that hold the bottom pan to the pedestal and carefully pull the bottom pan from the FRONT of the pedestal as if opening a book. Be careful not to strain any cables.
4. The pot marked "R3" is for the +12v±0.6v adjustment and the pot marked "R11" is for the +5v±0.25v adjustment.
5. Replace the bottom pan. Be careful not to pinch any cables.

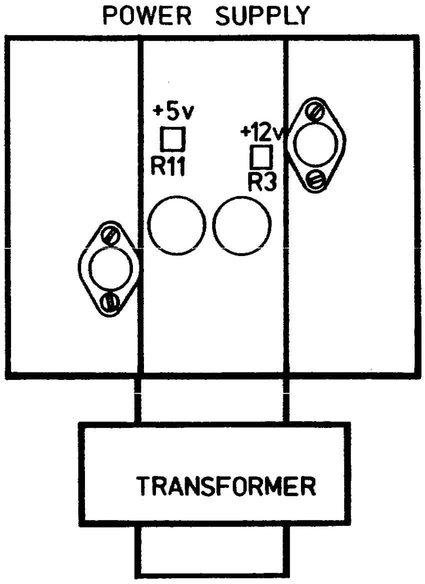


FIGURE 3-1

3.1 CRT DISPLAY ADJUSTMENTS

The VT4R, with the Elston Monitor, has nine potentiometers which affect the quality of the display. Two pots on the back of the pedestal control the contrast and brightness of the display. (A third pot in this location is for speaker volume.) These three pots are user accessable. The other seven pots are inside the CRT housing of the VT4R.

Use the following procedure to adjust the monitor display of the VT4R with the Elston Monitor:

1. Load the ATP called "ACRT4." A screen prompt will ask for a controller number. Enter the number of the controller of the VT4(s) being tested. A prompt will ask if a printer is to be used to display errors. Use Flag 2 to answer "yes" or Flag 3 to answer "no."

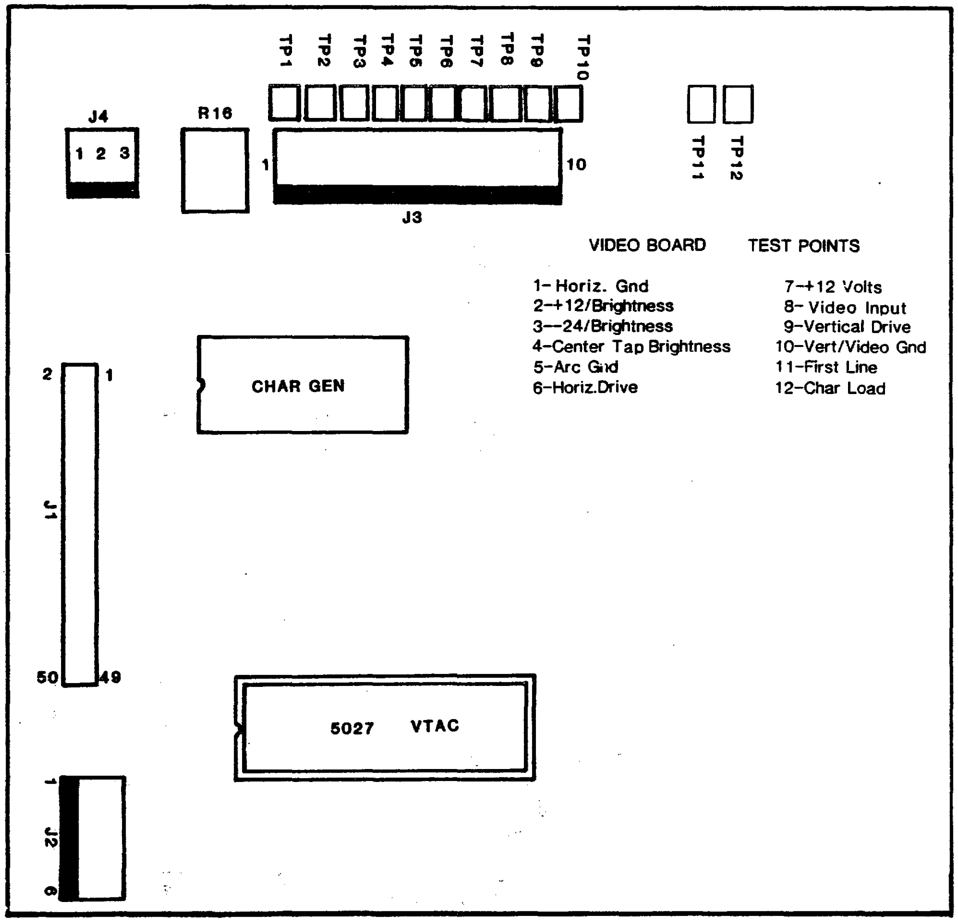
   An alignment pattern will be written to each available VT4. Varied forms and intensities of characters will be displayed on this pattern.

2. Adjust the brightness pot on the back of the pedestal to full intensity. Adjust the brightness pot on the Monitor Board to its maximum, and then turn it down until the raster just disappears.

3. Use shading pot R16 on the Video Board (Figure 3-2) to adjust the display so that the differences of each display line can be distinguished. These lines appear as follows:

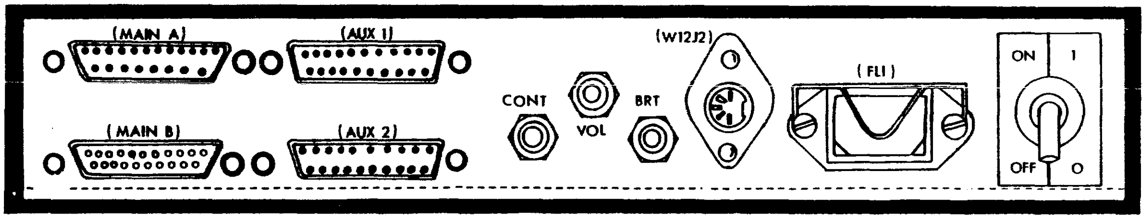
   Foreground
   Shaded Foreground
   Bold Foreground
   Background
   Shaded Background
   Bold Background

   The adjustment latitude that allows these different display features to all be distinguished is critical. Adjustment of the contrast and brightness potentiometers at the rear of the pedestal may need to be changed to make the subtle display differences perceptible.



VIDEO BOARD

Figure 3-2



REAR PANEL - VT4

Figure 3-3

4. Adjust the focus, size, and centering of the pattern as necessary with the pots on the Monitor Board (Figure 3-4). The adjustment tabs on the deflection yoke may also be used for positioning of the pattern.

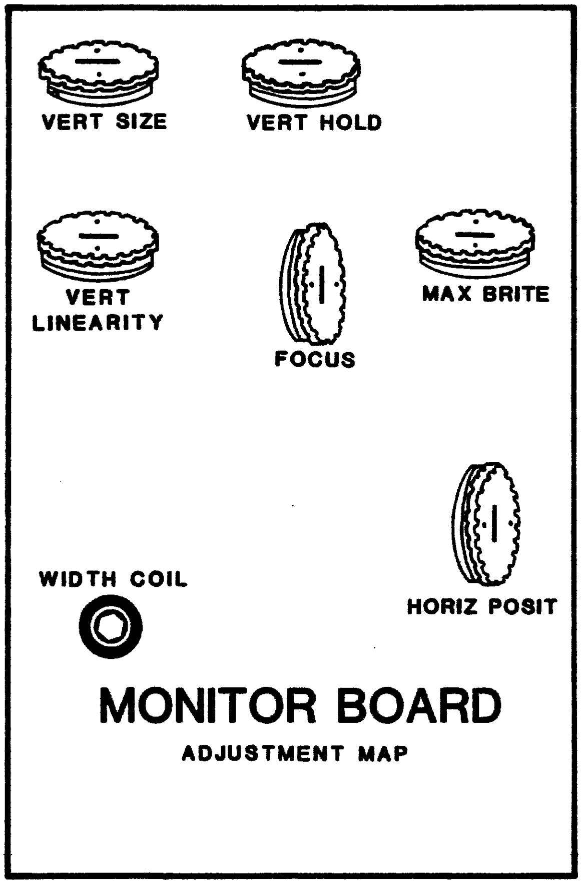


Figure 3-4

POWER MUST ALWAYS BE TURNED OFF BEFORE ATTEMPTING ANY REMOVALS OR REPLACEMENTS.

4.1 LOGIC BOARDS

4.1.1 CPU Board

The CPU Board is the large board that fits across the back of the CRT housing. It is held in place (upside down) by a standoff in each upper corner of the rear of the CRT housing and by a card guide on the rear of the CRT housing floor.

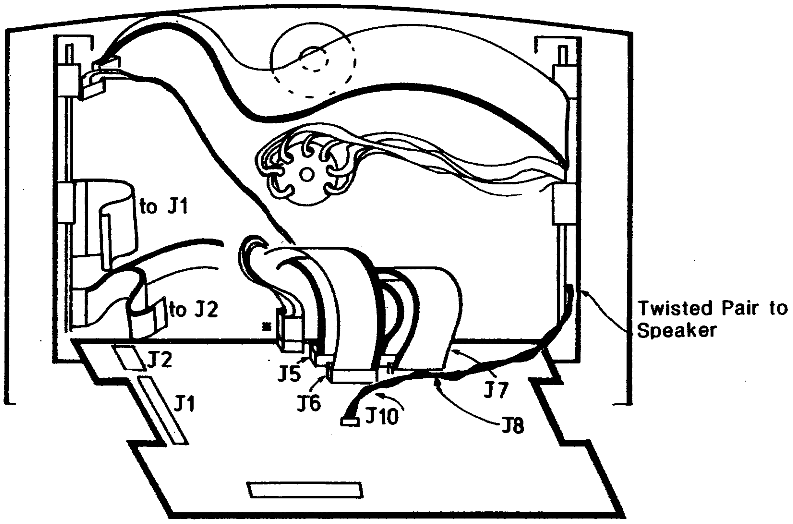


Figure 4-1

● Connector to the Control Panel (Brightness, Contrast)

Refer to Figure 4-1 and use the following procedure to remove the CPU Board:

1. Release the standoffs that fasten the bottom corners (which will be facing up) to the rear of the CRT housing.

2. Lift slightly on the board to free the top edge from the card guide.

3. Observe the locations and orientations of the eight cables that are connected to the board (refer to Figure 4-1).

4. Remove the CPU to Video Interface Cable from J1.

5. Remove the Power Cable from J2.

6. Remove the Main A Cable from J5.

7. Remove the Main B Cable from J6.

8. Remove the Aux 1 Cable from J7.

9. Remove the Aux 2 Cable from J8.

10. Remove the Control Panel Cable from J9.

11. Remove the Speaker Cable (twisted pair) from J10.

12. Reverse the above procedure to replace the CPU board.

4.1.2 Video Board

The Video Board is attached to the left (when viewed from the rear of the terminal) side of the CRT housing. The bottom of the Video Board rests in a card guide and the upper two corners are held in place by standoffs.

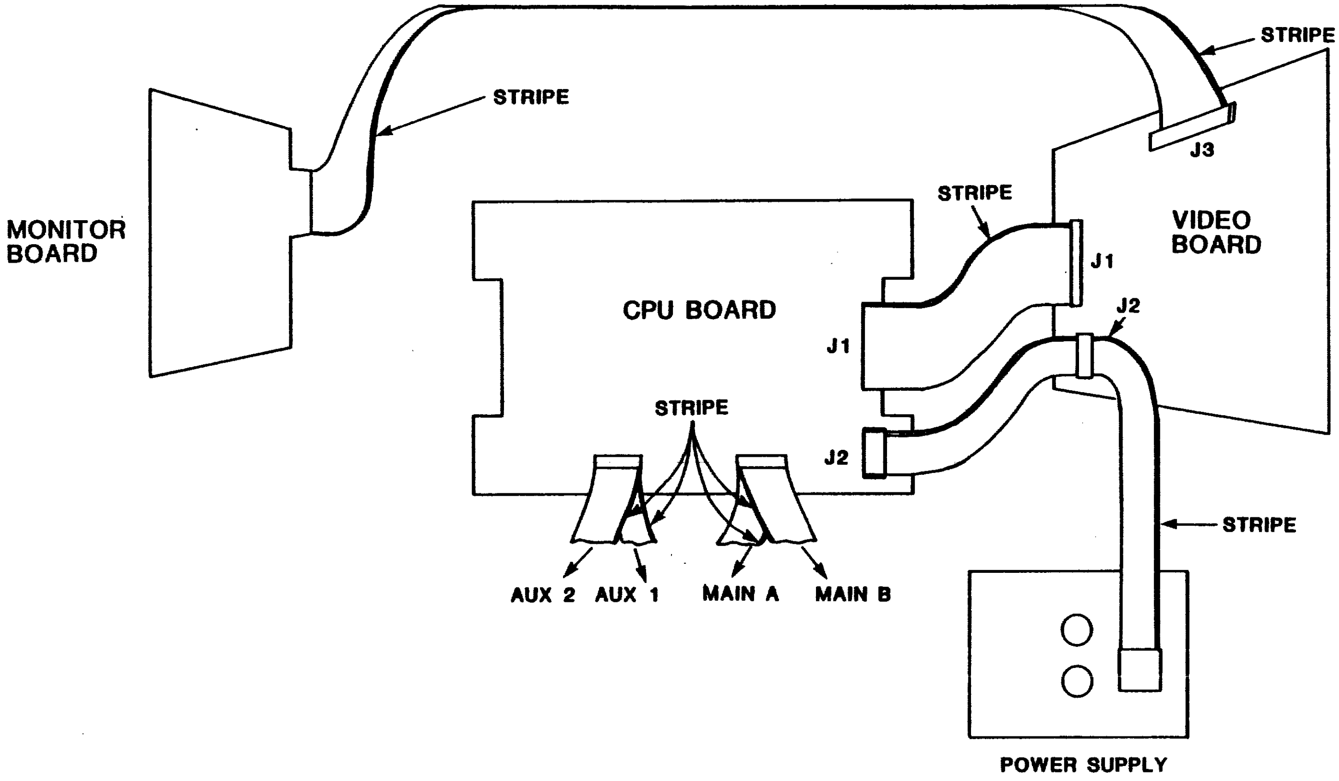


Figure 4-2

The Video Board is removed in the following manner:

1. Take note of the locations and orientations of the four cable connectors that attach to the board.

2. Remove the 50-conductor CPU to Video Interface Cable from J1 on the Video Board (the stripe faces up).

3. Remove the Power Cable from J2 on the Video Board (the stripe faces up).

4. Remove the Video to Monitor Interconnection Cable from J3 on the Video Board (the stripe goes towards the rear of the terminal at the J3 end).

5. Remove the connector with the twisted trio of wires from J4 (the blue wire goes towards the rear of the terminal).

6. Release the two standoffs and lift the board free.

7. Reverse the above procedure to replace the Video Board.

4.1.3 Monitor Board

Use the following procedure to remove the Monitor Board:

1. Detach the two black wires with the tab connectors from the neck of the tube.

2. Detach the connector from the base of the tube.

3. Detach the 6-wire Flyback Transformer Assembly Connector from the Monitor Board.

4. Detach the Bleeder Resister wire from the grounding pin on the Monitor Board.

5. Detach the 4-wire Deflection Yoke Connector from the Monitor Board.

6. Detach the Video to Monitor Interconnection Cable from the Monitor Board (the stripe faces down at the Monitor Board end of the cable).

7. Release the standoffs.

8. Reverse the procedure to replace the Monitor Board.

4.2 POWER SUPPLY

4.2.1 Power Supply PWA

Follow these steps to remove the PWA:

1. Tilt the VT4 on its back.

2. Remove the seven Phillips Screws and open the bottom pan. Be careful not to strain any of the internal cables.

3. Disconnect the two Molex connectors from the PWA.

4. Remove the four screws with a 1/4-inch nutdriver.

5. Reverse the above procedure to replace the Power Supply PWA.

4.2.2 Power Supply Transformer

Follow these steps to remove the transformer:

1. Tilt the terminal on its back.

2. Remove the screws that fasten the bottom pan to the pedestal. Be careful not to strain any of the internal cables when opening the pan.

3. Remove the cover from the terminal block and remove the transformer wires from pins 2 through 8.

4. Use a 5/16-inch nutdriver to remove the four screws that hold down the transformer.

5. Reverse the above procedure to replace the transformer.

   The wires from the transformer attach to the terminal block in this sequence:

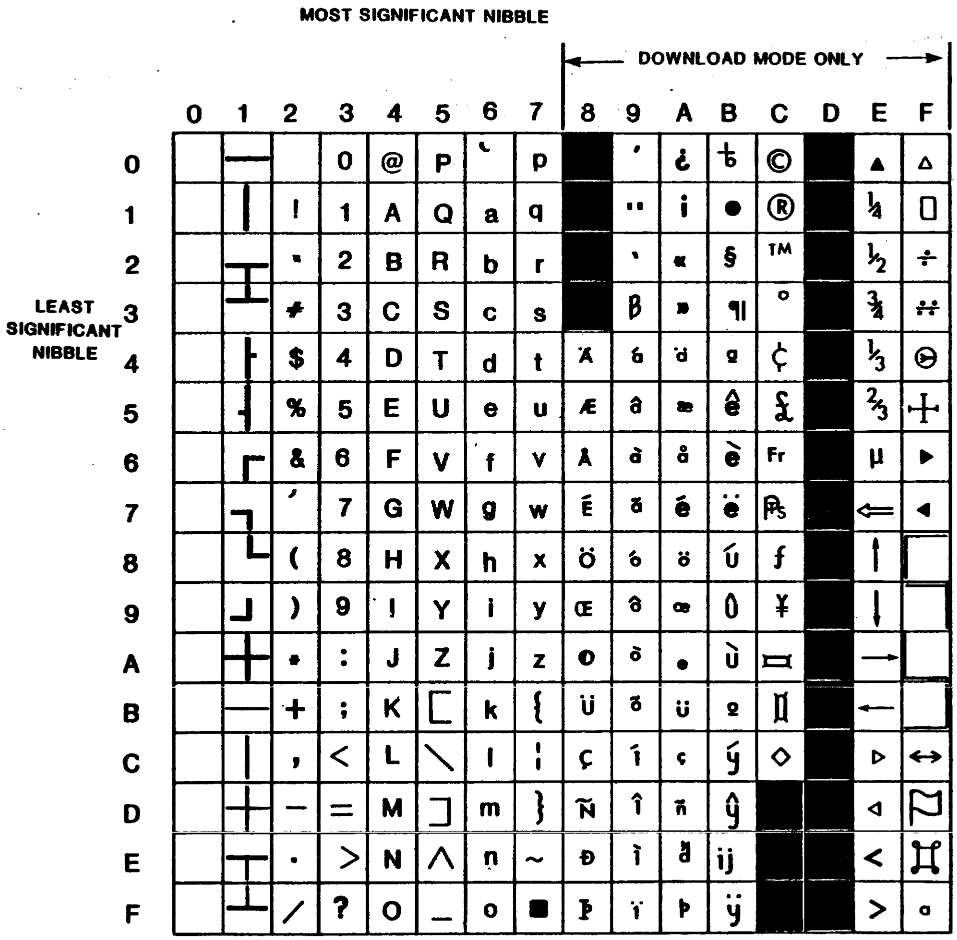
10 -		-
9 -		-
8 -	Orange -	125 VAC
7 -	Violet/White -	115 VAC
6 -	Red/White -	100 VAC
5 -	Brown/White -	0 VAC
4 -	Violet -	115 VAC
3 -	Red -	100 VAC
2 -	Brown -	0 VAC
1 -		-

When the VT4 is powered on, it emulates a VT3. The BEST Operating System is unable to distinguish between a VT3 and a VT4 unless the operator has configured the terminal as type 5 in the BEST CFIG table. The terminal will operate the same whether it is configured in CFIG as a VT3 (type 4) or as a VT4 (type 5).

5.1 CHARACTER SET

In VT3 Mode the VT4 character set is the same as that of the VT3 except that the Greek characters (hex 84 - 8F) are not present. Styles of some characters may vary slightly when the VT4 display is compared to a VT3 display. In VT3 mode, the characters corresponding to hex 10 - 7F are repeated when the terminal receives hex 90 - FF.

In VT4 Mode the character set is extended to another 128 characters (hex 80 - FF). However, hex 80 - 84 and hex CD - DF all cause a completely filled character dot matrix to be displayed (see Figure 5 - 1).



MOST SIGNIFICANT NIBBLE

DOWNLOAD MODE ONLY (8 - F)

LEAST SIGNIFICANT NIBBLE

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		─		0	@	P	`	p	█	´	¿	ƀ	©	█	▲	△
1		│	!	1	A	Q	a	q	█	¨	¡	●	®	█	¼	▯
2		┬	"	2	B	R	b	r	█	`	«	§	™	█	½	÷
3		┴	#	3	C	S	c	s	█	β	»	¶	°	█	¾	∺
4		├	$	4	D	T	d	t	Ä	á	ä	º	¢	█	⅓	⊝
5		┤	%	5	E	U	e	u	Æ	â	æ	ê	£	█	⅔	✛
6		┌	&	6	F	V	f	v	Å	à	å	è	Fr	█	µ	▶
7		┐	'	7	G	W	g	w	É	ă	é	ë	₧	█	⇐	◀
8		└	(	8	H	X	h	x	Ö	ó	ö	Ú	ƒ	█	↑	┌
9		┘	)	9	I	Y	i	y	Œ	ô	œ	Û	¥	█	↓	┐
A		┼	*	:	J	Z	j	z	◐	ò	•	Ù	⌑	█	→	└
B		─	+	;	K	[	k	{	Ü	ŏ	ü	º	⌷	█	←	┘
C		│	,	<	L	\	l	¦	Ç	í	ç	ý	◇	█	▷	↔
D		┼	−	=	M	]	m	}	Ñ	î	ñ	ŷ	█	█	◁	⚐
E		┬	.	>	N	^	n	~	Đ	ì	ǎ	ĳ	█	█	<	⌘
F		┴	/	?	O	_	o	■	Þ	ï	þ	ÿ	█	█	>	°

VT4R
CHARACTER SET

Figure 5-1

5.2 VT4 CONTROL CODES

The VT4 uses all the VT3 control codes (except 0E0F) as well as some control codes unique to the VT4 which are marked by an astrisk (*) in the list below:

CODE

00 Null
Ignored, except causes write initialization.

01xxyy Set Cursor
Causes the terminal to place the cursor at the Row xx, Column yy; xx between 1 and 27 (24); yy between 0 and 63 (79).

04 Escape
The next character is a control character.

040000* Return to VT3 Mode
Allows the user to return to VT3 Mode.

040001* Extended Status Request
The VT4 will drop any pending Read Requests and will prepare the following message:

FFFF - dev type - rev level - load id - error - 303030303030 . . . (up to 22 bytes, depending on the program).

After posting the above message the VT4 will set Read Request. If devtype = 11, the CPM ROM is present. If devtype = 01, the CPM ROM is not present.

040002* 1 Key Transmit
The next displayable character generated by the operator will be displayed and transmitted.

0401 Remember Cursor
Saves the current value of the cursor location for the Restore Cursor Command (09).

0402 Select 64 Character Mode
Sets the display mode to 64 characters by 27 lines. The screen is cleared to blanks and the cursor is positioned to home (or the 27th line if in Typewriter Mode.

0403 Select 80 Character Mode
Sets the display mode to 80 characters by 24 lines. The screen is cleared to blanks and the cursor is positioned to home (or the 24th line if in Typewriter Mode.

0404 Select Default
Sets the display mode to 64 characters per line

0405 Set Blind Entry Mode
Clears the 32 character blind buffer to blanks. At the completion of the write, the cursor is positioned to the beginning of the blind buffer.

0406 Blank Fill Line
Replaces all characters from the cursor position to the end of the line with blanks (foreground, background, or suppressed background). The cursor is then positioned to the beginning of the next line (or home).

0407 Clear Field
Replaces all characters in the field where the cursor resides with foreground blanks. The cursor is then positioned to the first entry in the field.

0408 Force Transmit
Causes the VT4 to set Read Request. The VT4 acts as if the operator pressed Transmit.

0409 Transmit Stop
Stored on the screen as a suppressed background character. When the operator presses TAB or RETURN and the cursor advances past a Transmit Stop, the VT4 posts Read Request. In addition SHIFT/TAB and SHIFT/RETURN will allow the cursor to back up over a Transmit Stop Character.

040A Roll Up
Moves the screen portion below the cursor line up one line. The cursor line is lost. A blank foreground line is inserted at the bottom of the screen. The cursor is positioned at the beginning of the bottom line. The control line is not affected.

040B Roll Down
Moves the screen portion below and including the cursor line down one line. A blank foreground line is inserted at the cursor line. The cursor is positioned at the beginning of the current line. The control line is not affected.

040C Set Suppressed Background
Causes the terminal to accept subsequent data characters as suppressed background characters (displayed as blanks).

05 Clear Screen
Clears the screen to foreground blanks and positions the cursor at the home position. The control line is not affected.

06 Clear Foreground
Clears all foreground positions on the screen to blanks and positions the cursor at the first foreground position on the screen. The control line is not affected.

07 Alarm
Causes the audible alarm to sound.

08 Background Follows
Causes the terminal to accept subsequent characters as background (displayed as black on green) characters.

09 Restore Cursor
Positions the cursor to the position it occupied at the start of the write or to the last 'remembered' cursor position (0401).

0A Foreground Follows
Causes the terminal to accept subsequent characters as foreground (displayed as green on black) characters.

0B Right-Justified Field
When placed as the character immediately preceding a foreground field, this character designates the field to be right-justified. This character is stored on the screen as a suppressed background character.

0C Transmit Mark
Stored on the screen as a suppressed background character. When the operator presses TAB or RETURN and the cursor advances past a Transmit Mark (also called Transmit Delimiter), the VT4 posts a Read Request.

0D Carriage Return
Causes the cursor to be positioned at the beginning of the next line down.

0E Escape
The next character is a control character.

0E02 Reset F2
Resets Flag 2.

0E03 Reset F3
Resets Flag 3.

0E04 Reset F2/F3
Resets Flags 2 and 3.

0E05 Enter Typewriter Mode
Sets Typewriter Mode and clears Normal Mode.

0E06 Enter Normal Mode
Sets Normal Mode and clears Typewriter Mode.

0E07 Write Control Line
Clears the Control Line to blanks. Any subsequent data is written to the Control Line.

0E08* Download to RAM
The data following this instruction will be downloaded to RAM. See section 5.5 for details on downloading Z80 Microcode to RAM.

0E09* Execute RAM Code
The data following this instruction is the two byte address of previously downloaded Z80 Microcode residing in RAM.

0E0A Keyboard Data Follows
The following data will be treated as if it were entered from the keyboard.

0E0B* Reserved
Reserved sequence to flag data intended for special QSP devices.

0E0F Execute Test Program
This is illegal on a VT4. This instruction and all subsequent data in the write will be ignored.

0F Kana Field
This is ignored by the VT4 except as a placeholder.

5.3 MEMORY ORGANIZATION

The VT4 contains 12K of ROM and 64K of RAM. The 12K of ROM overlaps the first 12K of RAM. Once the terminal has been initialized, it is possible to load Z80 code to any location in RAM. However, some address locations should be avoided. Code should not be loaded into hex addresses above E000 until after the program has control of the Z80. The VT4 will not function correctly if hex addresses F800 - FFFF are overwritten as these addresses contain QSP Buffers, Interrupt Tables, Interface Variables, and Internal Interface Codes.

ROM	
$0000-$1FFF (8K)	VT3 Emulation & Interrupt Handlers
$2000-$2FFF	CPM ROM

RAM	
$0000-$00FF	Interface (CPM)
$0100-$C7FF (50K)	User Area (TPA - Transient Program Area)
$C800-$CFFF (2K)	CCP (CPM)
$D000-$DFFF (4K)	BDOS (Disc File System for CPM)
$E000-$E7FF (2K)	BIOS (I/O for CPM, VT3-emulation Variables
$E800-$F7FF (4K)	CRT Display Screen Area for VT3-emulation, CPM
$F800-$FFFF (2K)	QSP Buffers, Interrupt Tables, Interface Variables Int. Interface Code

Table 5-1

SPECIFIC RAM VARIABLES AND INTERRUPT HANDLERS

MNEMONIC	ADDRESS	LENGTH	DESCRIPTION
	$F800	2	Port D Interrupt, Transmit Buffer Empty
	$F802	2	Port D Interrupt, External Status Change
	$F804	2	Port D Interrupt, Receive Character Available
	$F806	2	Port D Interrupt, Special Receive Conditions
	$F808	2	Port C Interrupt, Transmit Buffer Empty
	$F80A	2	Port C Interrupt, External Status Change
	$F80C	2	Port C Interrupt, Receive Character Available
	$F80E	2	Port C interrupt, Special Receive Conditions
	$F810	2	Port B Interrupt, Transmit Buffer Empty
	$F812	2	Port B Interrupt, External Status Change
	$F814	2	Port B Interrupt, Receive Character Available
	$F816	2	Port B Interrupt, Special Receive Conditions
	$F818	2	Port A Interrupt, Transmit Buffer Empty
	$F81A	2	Port A Interrupt, External Status Change
	$F81C	2	Port A Interrupt, Receive Character Available
	$F81E	2	Port A Interrupt, Special Receive Conditions
CTC3	$F826	2	CTC Channel 3, Video Refresh Interrupt
TERM	$F828	1	QSP Terminal Address
STATUS	$F829	1	QSP Status Byte $80 Reserved $40 Flag 3 $20 Flag 2 $10 Read Pending $08 Reserved $04 Reserved $02 Write Pending (Busy) $01 Reserved
TBUFADR	$F82A	2	QSP Transmit Buffer Address
TBUFLEN	$F82C	2	QSP Transmit Buffer Length (0-$320)
RBUFADR	$F82E	2	QSP Receive Buffer Address
RBUFLEN	$F830	2	QSP Receive Buffer Length
	$F832	$3C	Reserved for QSP
	$F86E	2	Address to return to VT3 emulation
SCANST	$F870	2	Video Refresh Table Start Address
	$F872	2	Reserved for Video Refresh Routine
RBUFFER	$F900	$320	QSP Receive Buffer
TBUFFER	$FC40	$320	QSP Transmit Buffer

Table 5-2

5.4 I/O TABLE

The following table lists the various I/O ports of the VT4:

PORT	DESCRIPTION
00 - 07	Test Panel (DPP1)
08 & 0C	SIO-A data (QSP)
09 & 1D	SIO-A control
0A & 0E	SIO-B data (AUX 1)
0B & 0F	SIO-B control
10 & 14	CTC-0 (speaker frequency)
11 & 15	CTC-1 (keyboard serial clock)
12 & 16	CTC-2 ((AUX 2 serial clock)
13 & 17	CTC-3 (video interrupt)
18 & 1C	SIO-C data (AUX 2)
19 & 1D	SIO-C control
1A & 1E	SIO-D data (keyboard)
1B & 1F	SIO-D control
20 - 27	Baud Rate Generator
28 - 2F	JMP 6 (Device B and Address B select)
30 - 37	JMP 7 (50/60 Hz, Poll/Test, 64 Character Expansion)
38 - 3F	Parity Clear and RAM/ROM Switch
40 - 47	JMP 4 (AUX 1 baud rate)
48 - 4F	Line Attribute
50 - 57	Parity and RAM/ROM Readback
58 - 5F	DMA count and Start DMA
60 - 67	Character Buffer Offset
68 - 6F	DMA Address (LSB)
70 - 77	DMA Address (MSB)
78 - 7F	Character Mode Switch
80 - 87	VTAC Register 0 (horizontal line count)
88 - 8F	VTAC Register 1 (horizontal synch delay)
90 - 97	VTAC Register 2 (scans/data and characters/data row)
98 - 9F	VTAC Register 3 (skew and data rows/frame)
A0 - A7	VTAC Register 4 (scan lines/frame)
A8 - AF	VTAC Register 5 (vertical synch delay)
B0 - B7	VTAC Register 6 (last displayed data row)
B8 - BF	VTAC (self-load - not used)
C0 - C7	VTAC (cursor line address - not used)
C8 - CF	VTAC (cursor position address - not used)
D0 - D7	VTAC (reset)
D8 - DF	VTAC (upscroll - not used)
E0 - E7	VTAC (load cursor character position - not used)
E8 - EF	VTAC (load cursor line address - not used)
F0 - F7	VTAC (start timing chain)
F8 - FF	VTAC (non-processor self-load - not used)

Table 5-3

5.5 DOWNLOADING MICROCODE

The VT4 is an intelligent terminal and is capable of running Z80 programs independantly of the host. This capability is primarily designed to be used for word processing. When the terminal is in the word processing (*QICWORD) mode, special display parameters are used that are not available when the terminal is acting as a VT3.

The user can download Z80 machine code to any location in RAM. The downloading sequence is as follows:

```
                    0E08aaaannnndddd . . . dddd
```

Where aaaa is the starting address of the code, nnnn is the number of bytes to be loaded, and dddd . . . dddd is the actual machine code.

After the code is downloaded, the terminal must receive the instruction to execute the code. This requires a separate write to the terminal:

```
                              0E09aaaa
```

Where aaaa is the starting address of the code to be executed.

5.6 GENERAL MICROCODE ROUTINES

When the VT4 is first powered on, it runs through a pre-initialization, a self-test, a full initialization, and then it falls into the Main Idle (Control) Loop.

The Main Control Loop and its various subroutines checks the keyboard, Write Pending Flag, and Read Pending Flag. The following flow charts offer a very general illustration of the program flow in the VT4:

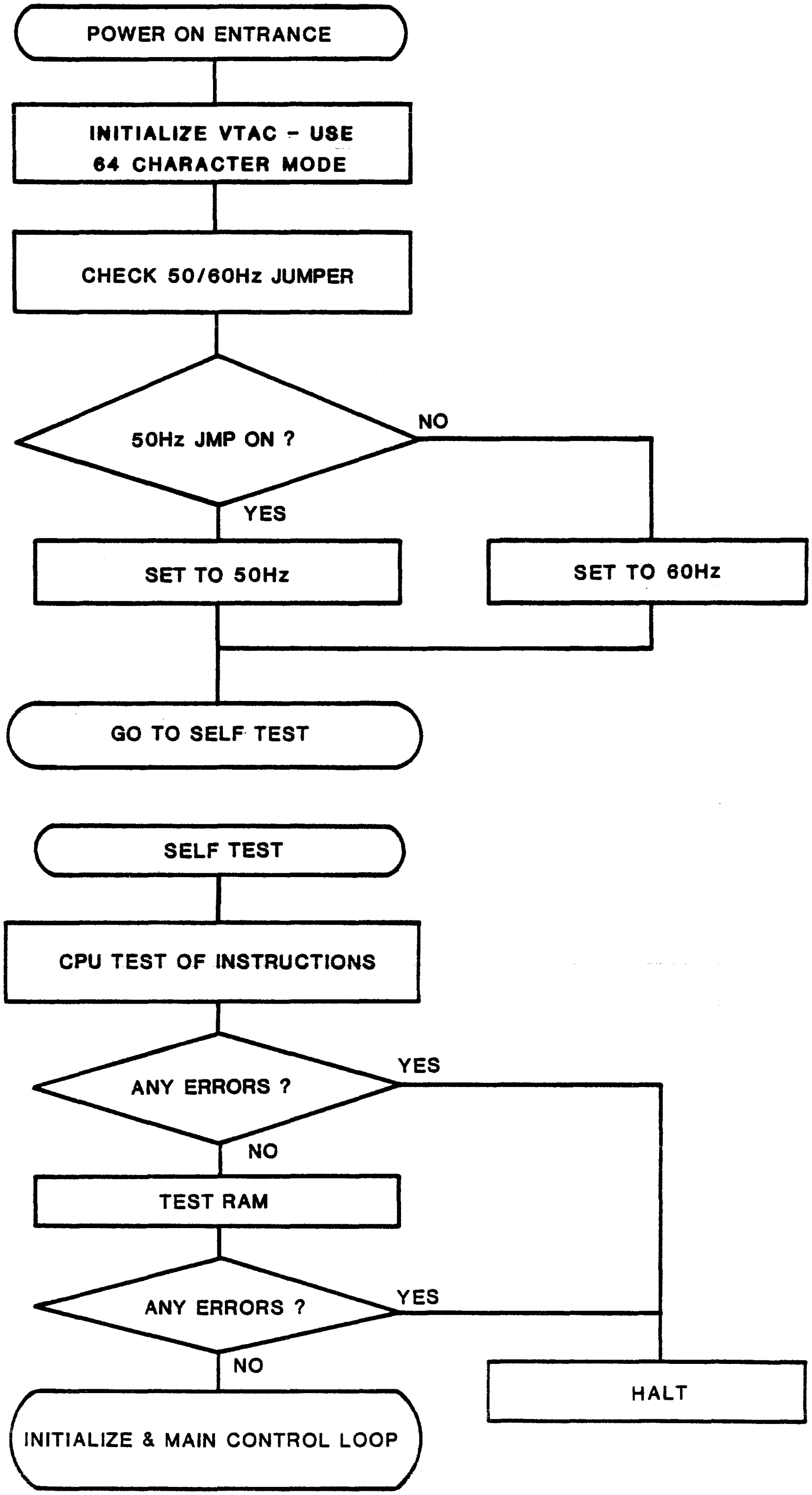

POWER ON ENTRANCE
INITIALIZE VTAC - USE 64 CHARACTER MODE
CHECK 50/60Hz JUMPER
50Hz JMP ON ?
NO
YES
SET TO 50Hz
SET TO 60Hz
GO TO SELF TEST
SELF TEST
CPU TEST OF INSTRUCTIONS
ANY ERRORS ?
YES
NO
TEST RAM
ANY ERRORS ?
YES
NO
HALT
INITIALIZE & MAIN CONTROL LOOP

Figure 5-2

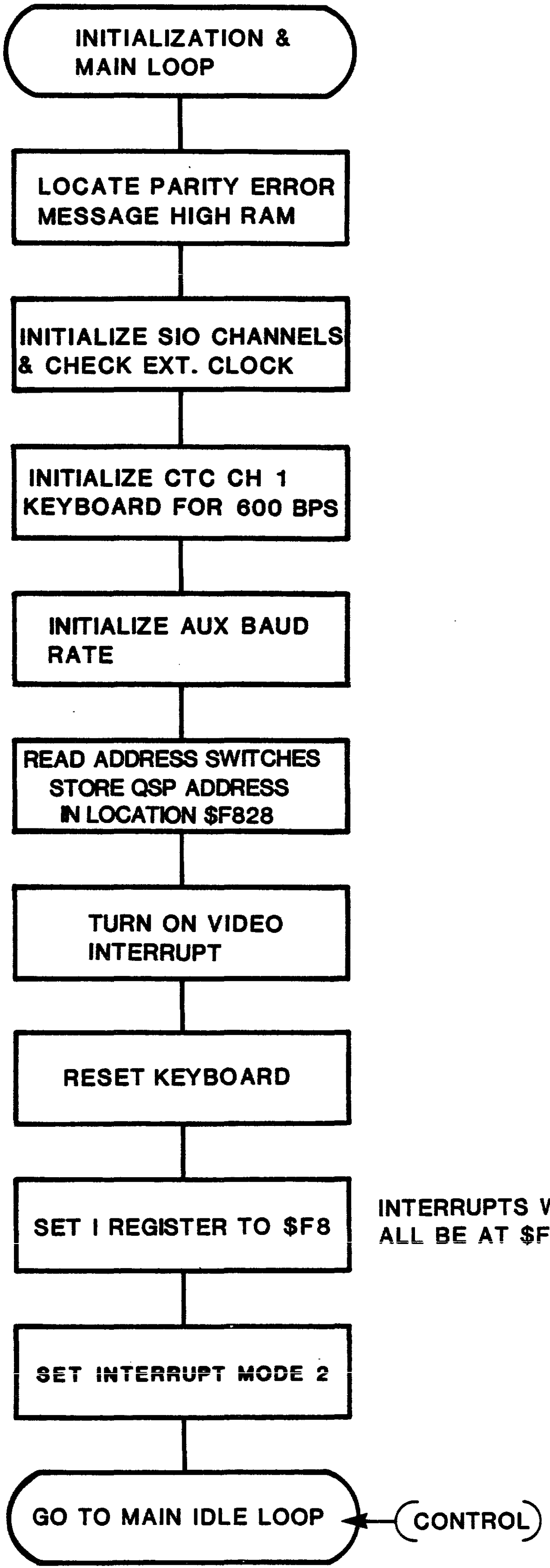

INITIALIZATION & MAIN LOOP
LOCATE PARITY ERROR MESSAGE HIGH RAM
INITIALIZE SIO CHANNELS & CHECK EXT. CLOCK
INITIALIZE CTC CH 1 KEYBOARD FOR 600 BPS
INITIALIZE AUX BAUD RATE
READ ADDRESS SWITCHES STORE QSP ADDRESS IN LOCATION $F828
TURN ON VIDEO INTERRUPT
RESET KEYBOARD
SET I REGISTER TO $F8
INTERRUPTS WILL ALL BE AT $F8XX
SET INTERRUPT MODE 2
GO TO MAIN IDLE LOOP
CONTROL

Figure 5-3

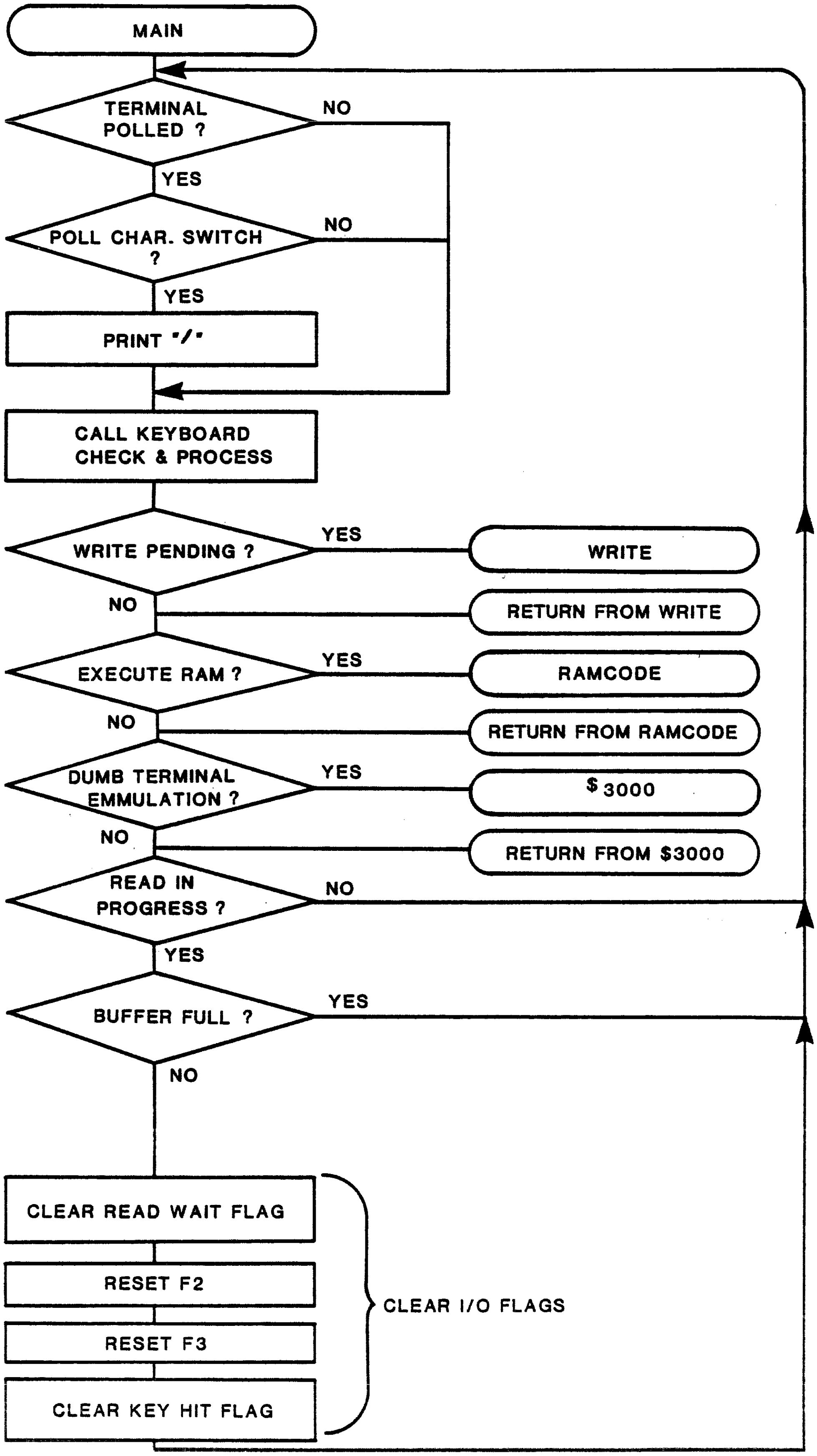

MAIN
TERMINAL POLLED ?
NO
YES
POLL CHAR. SWITCH ?
NO
YES
PRINT "/"
CALL KEYBOARD CHECK & PROCESS
WRITE PENDING ?
YES
WRITE
NO
RETURN FROM WRITE
EXECUTE RAM ?
YES
RAMCODE
NO
RETURN FROM RAMCODE
DUMB TERMINAL EMMULATION ?
YES
$ 3000
NO
RETURN FROM $3000
READ IN PROGRESS ?
NO
YES
BUFFER FULL ?
YES
NO
CLEAR READ WAIT FLAG
RESET F2
RESET F3
CLEAR KEY HIT FLAG
CLEAR I/O FLAGS

Figure 5-4

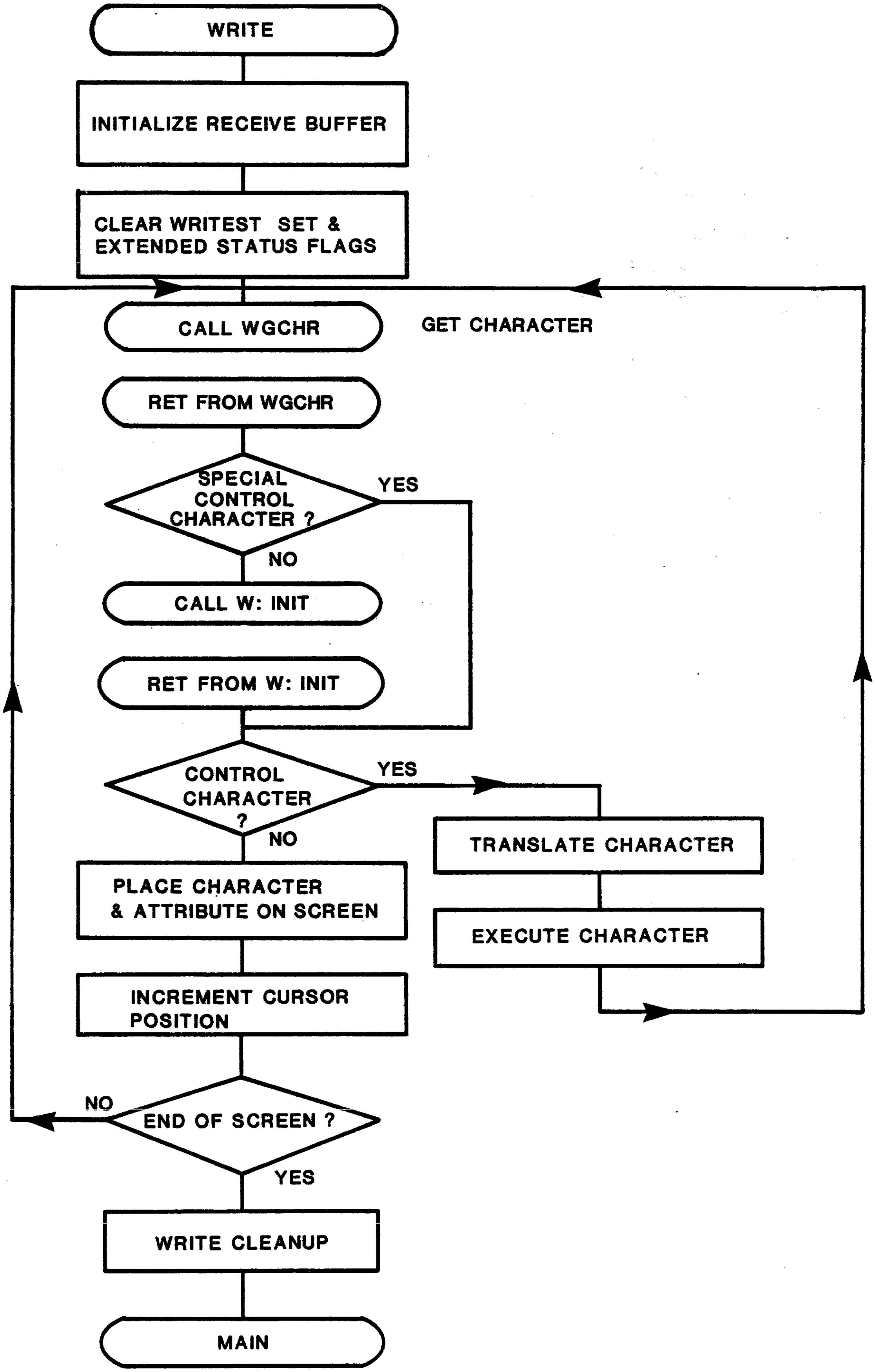

WRITE
INITIALIZE RECEIVE BUFFER
CLEAR WRITEST SET & EXTENDED STATUS FLAGS
CALL WGCHR
GET CHARACTER
RET FROM WGCHR
SPECIAL CONTROL CHARACTER ?
YES
NO
CALL W: INIT
RET FROM W: INIT
CONTROL CHARACTER ?
YES
NO
TRANSLATE CHARACTER
EXECUTE CHARACTER
PLACE CHARACTER & ATTRIBUTE ON SCREEN
INCREMENT CURSOR POSITION
NO
END OF SCREEN ?
YES
WRITE CLEANUP
MAIN

Figure 5-5

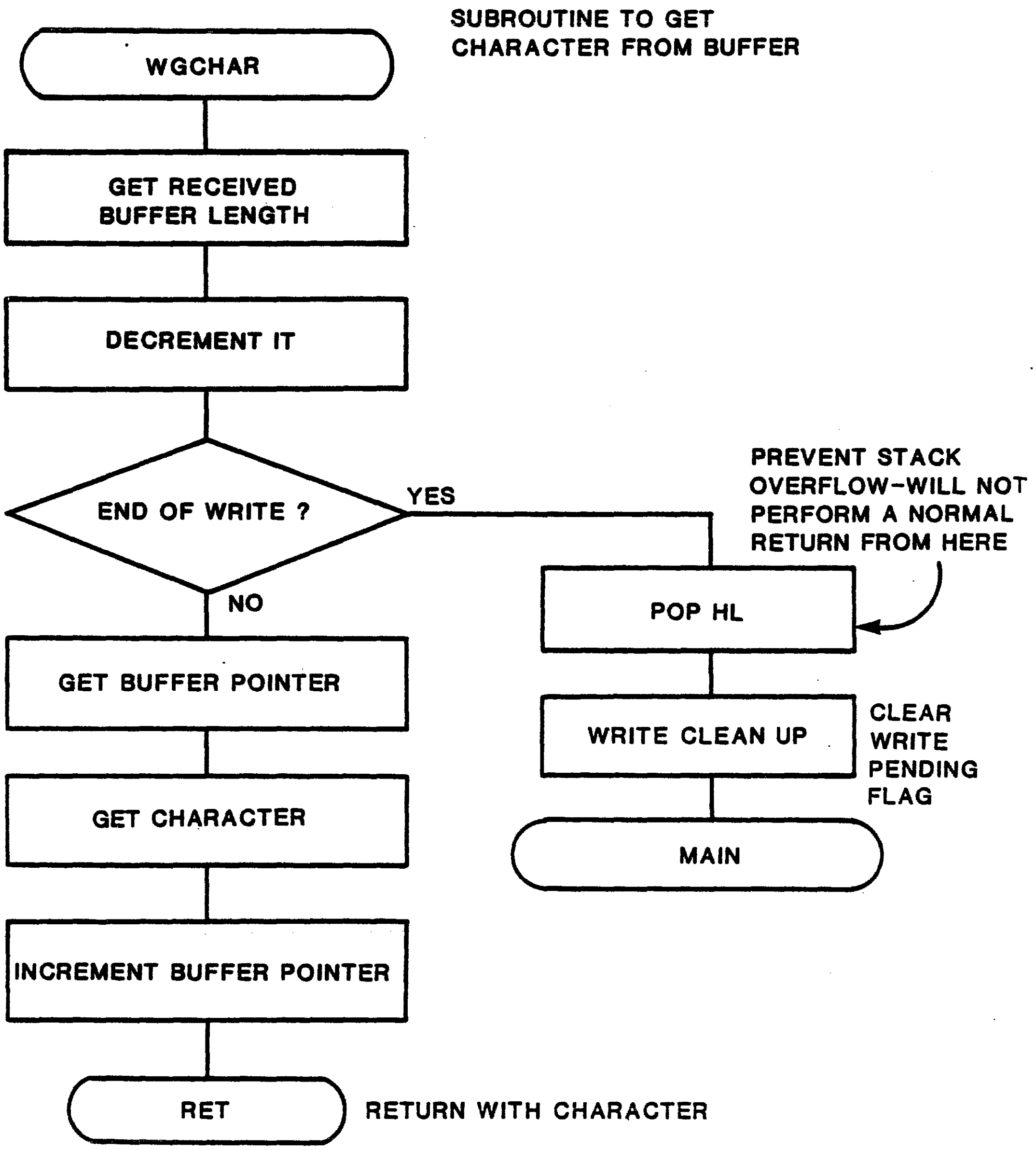

SUBROUTINE TO GET CHARACTER FROM BUFFER
WGCHAR
GET RECEIVED BUFFER LENGTH
DECREMENT IT
END OF WRITE ?
YES
NO
PREVENT STACK OVERFLOW-WILL NOT PERFORM A NORMAL RETURN FROM HERE
POP HL
WRITE CLEAN UP
CLEAR WRITE PENDING FLAG
MAIN
GET BUFFER POINTER
GET CHARACTER
INCREMENT BUFFER POINTER
RET
RETURN WITH CHARACTER

Figure 5-6

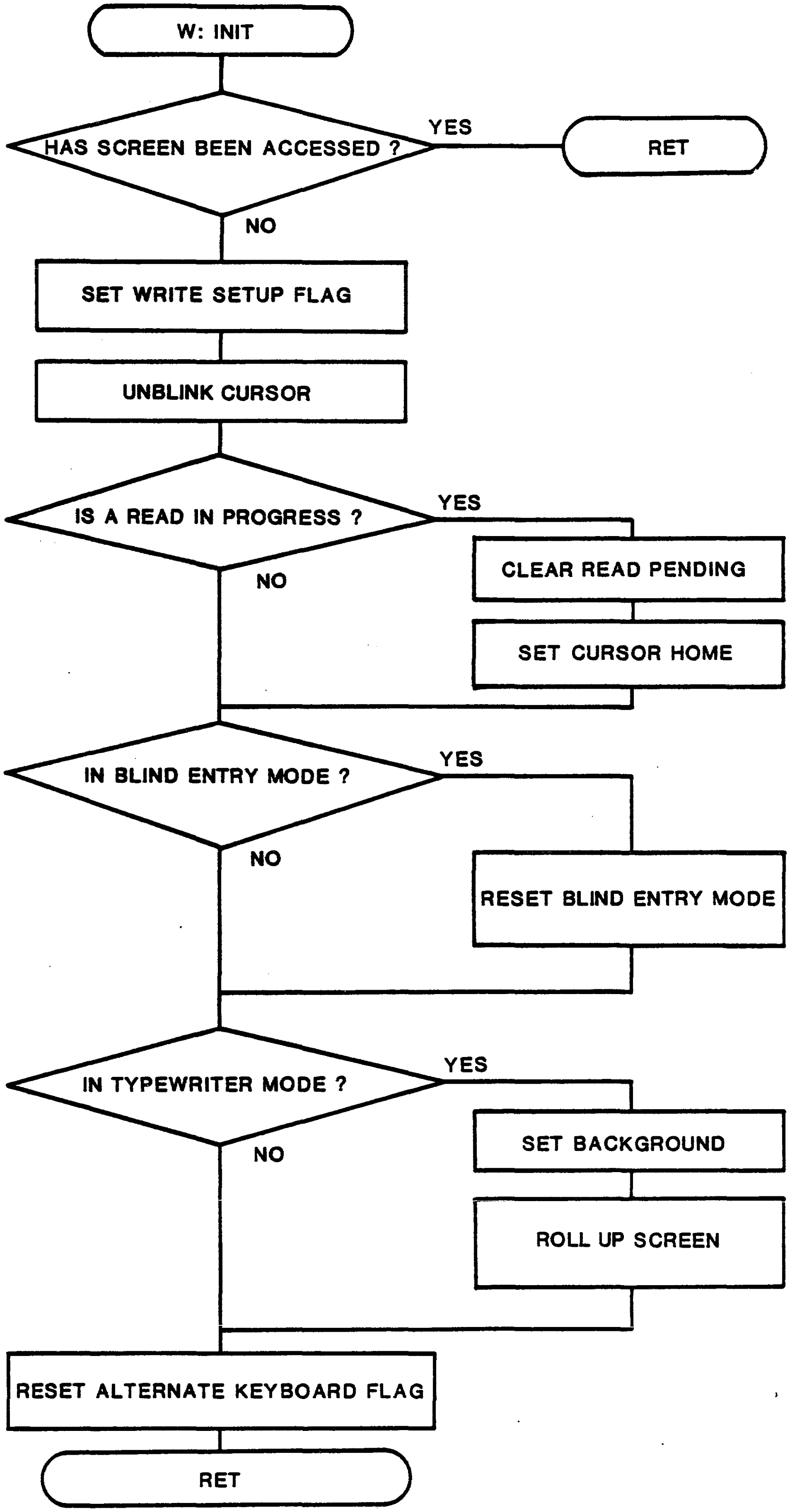

W: INIT
HAS SCREEN BEEN ACCESSED ?
YES
RET
NO
SET WRITE SETUP FLAG
UNBLINK CURSOR
IS A READ IN PROGRESS ?
YES
CLEAR READ PENDING
SET CURSOR HOME
NO
IN BLIND ENTRY MODE ?
YES
NO
RESET BLIND ENTRY MODE
IN TYPEWRITER MODE ?
YES
SET BACKGROUND
ROLL UP SCREEN
NO
RESET ALTERNATE KEYBOARD FLAG
RET

Figure 5-7

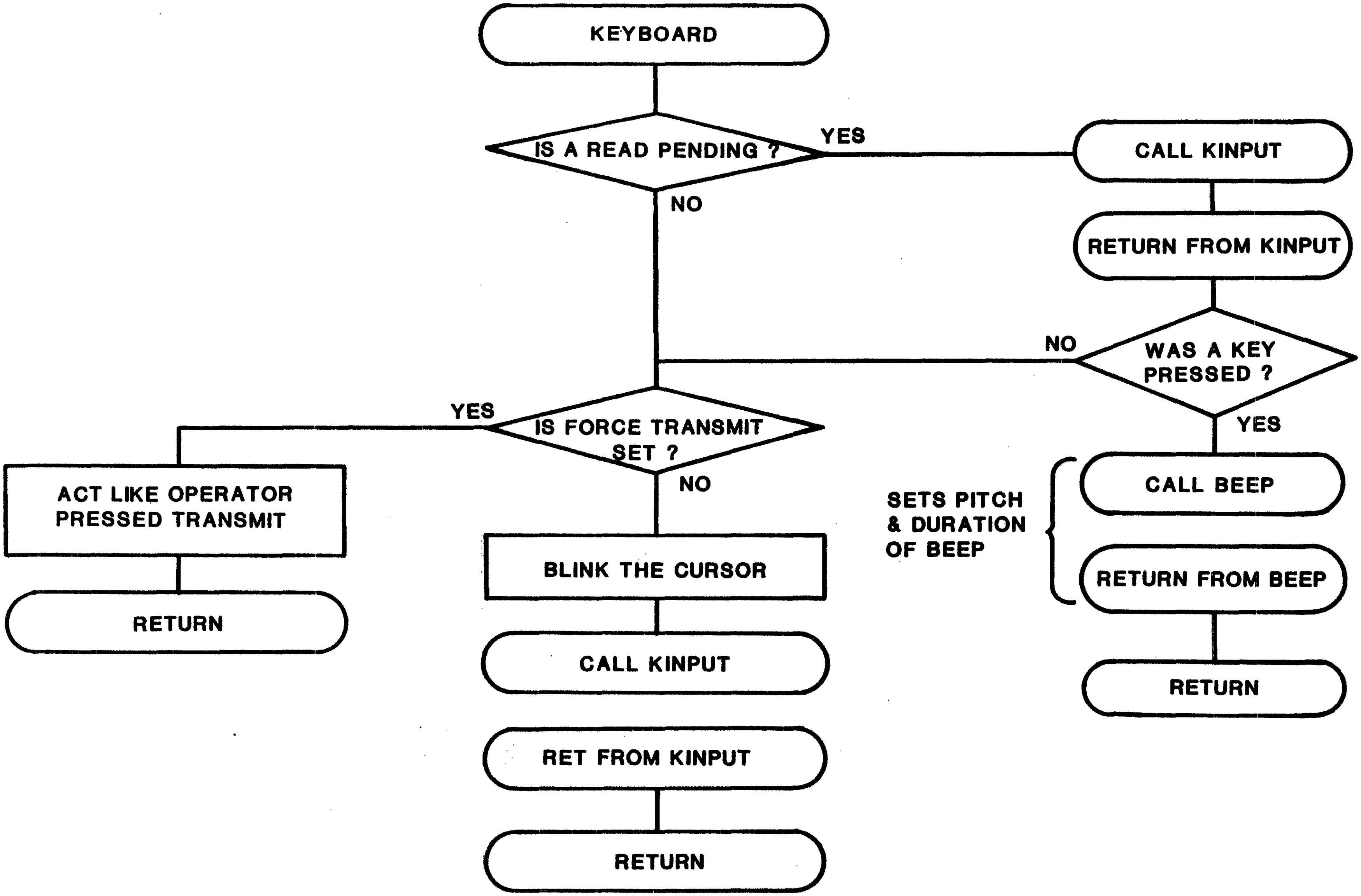

KEYBOARD
IS A READ PENDING ?
YES
NO
CALL KINPUT
RETURN FROM KINPUT
WAS A KEY PRESSED ?
NO
YES
IS FORCE TRANSMIT SET ?
YES
NO
ACT LIKE OPERATOR PRESSED TRANSMIT
RETURN
BLINK THE CURSOR
CALL KINPUT
RET FROM KINPUT
RETURN
SETS PITCH & DURATION OF BEEP
CALL BEEP
RETURN FROM BEEP
RETURN

Figure 5-8

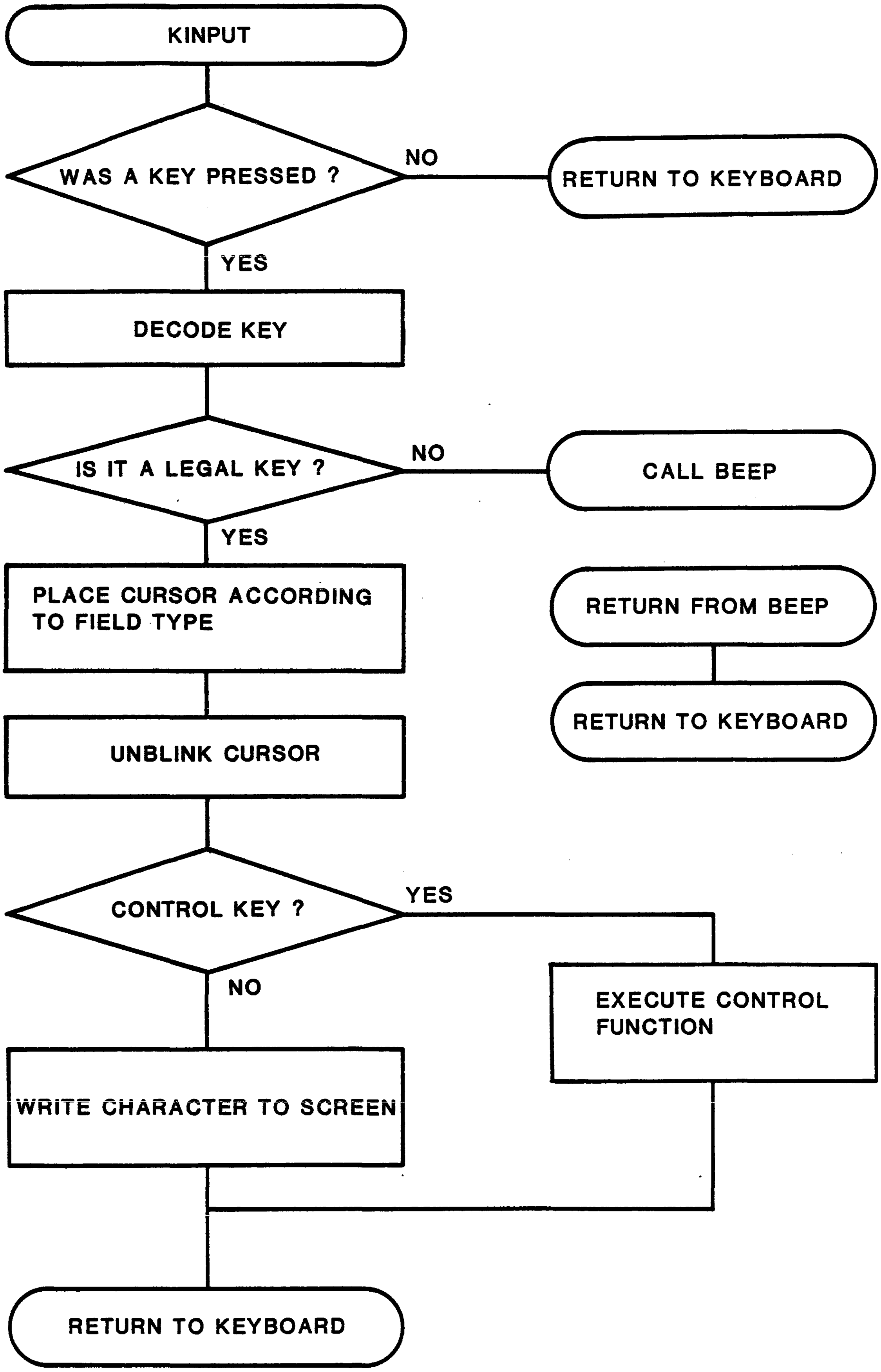

KINPUT
WAS A KEY PRESSED ?
NO
RETURN TO KEYBOARD
YES
DECODE KEY
IS IT A LEGAL KEY ?
NO
CALL BEEP
YES
PLACE CURSOR ACCORDING TO FIELD TYPE
RETURN FROM BEEP
RETURN TO KEYBOARD
UNBLINK CURSOR
CONTROL KEY ?
YES
NO
EXECUTE CONTROL FUNCTION
WRITE CHARACTER TO SCREEN
RETURN TO KEYBOARD

Figure 5-9

5.6.1 Keyboard Decoding

The KINPUT Subroutine does the decoding of the keyboard. The keyboard sends a key identification number to the terminal whenever a key is depressed. This key identification number is not a part of any standard code. Therefore, the microcode must determine the function of the key.

The microcode uses a key table with two bytes representing each key. The high byte represents the basic key function (e.g. alpha, numeric, special character, control, ten key, etc.). The low byte from the key table is the hexadecimal representation of the character that will be generated by the key. The non-controlling characters are standard ASCII characters.

The least significant bit sent back from the keyboard indicates whether or not the last keystroke was a downstroke or an upstroke (1 is up, 0 is down). If KINPUT does not detect a new key and the last detected keystroke was a downstroke, the microcode assumes that the key is being held down. The repeat routine is called in this case.

5.6.2 Control Characters

5.6.2.1 NON-RIGHT-JUSTIFIED MODE

The non-right-justified control characters are generated as follows:

BACKSPACE Moves the cursor backwards without overwriting characters.

SHIFT/BACKSPACE Moves the cursor forwards without overwriting characters.

RETURN Moves the cursor to the first available foreground field below the current cursor line. If a Transmit Delimiter is passed by the cursor, the program scans back to the beginning of the current foreground field, and Read Request is set.

SHIFT/RETURN Moves the cursor to the first available foreground field above the current cursor line.

CLEAR Clears the current foreground field.

SHIFT/CLEAR Clears all foreground characters from the screen.

TAB Moves the cursor to the first available foreground field to the right of or below the current foreground field. If a Transmit Delimiter is passed by the cursor, the program scans back to the beginning of the current foreground field and Read Request is set.

SHIFT/TAB Moves the cursor to the first available foreground field to the left of or above the current forefield.

INS Inserts a character space at the current cursor location and shifts all characters from the cursor forward, one space to the right. Any characters shifted beyond the end of the field will be truncated.

PAUSE Holds up Write Busy.

DEL (SHIFT/INS) Deletes the character upon which the cursor is resting and shifts all characters from the cursor forward, one space to the left.

TRANSMIT When the cursor rests between two Transmit Delimiters, all foreground characters are loaded into the Transmit Buffer and Read Request is set.

YES Enters 'Y' and a Force Transmit.

NO Enters 'N' and a Force Transmit.

5.6.2.2 RIGHT-JUSTIFIED MODE

The right-justified control characters are generated as follows:

BACKSPACE Removes the last character from the field and shifts the rest of the field one character position to the right.

RETURN Same as in non-right-justified mode.

SHIFT/RETURN Same as SHIFT/TAB in non-right-justified mode.

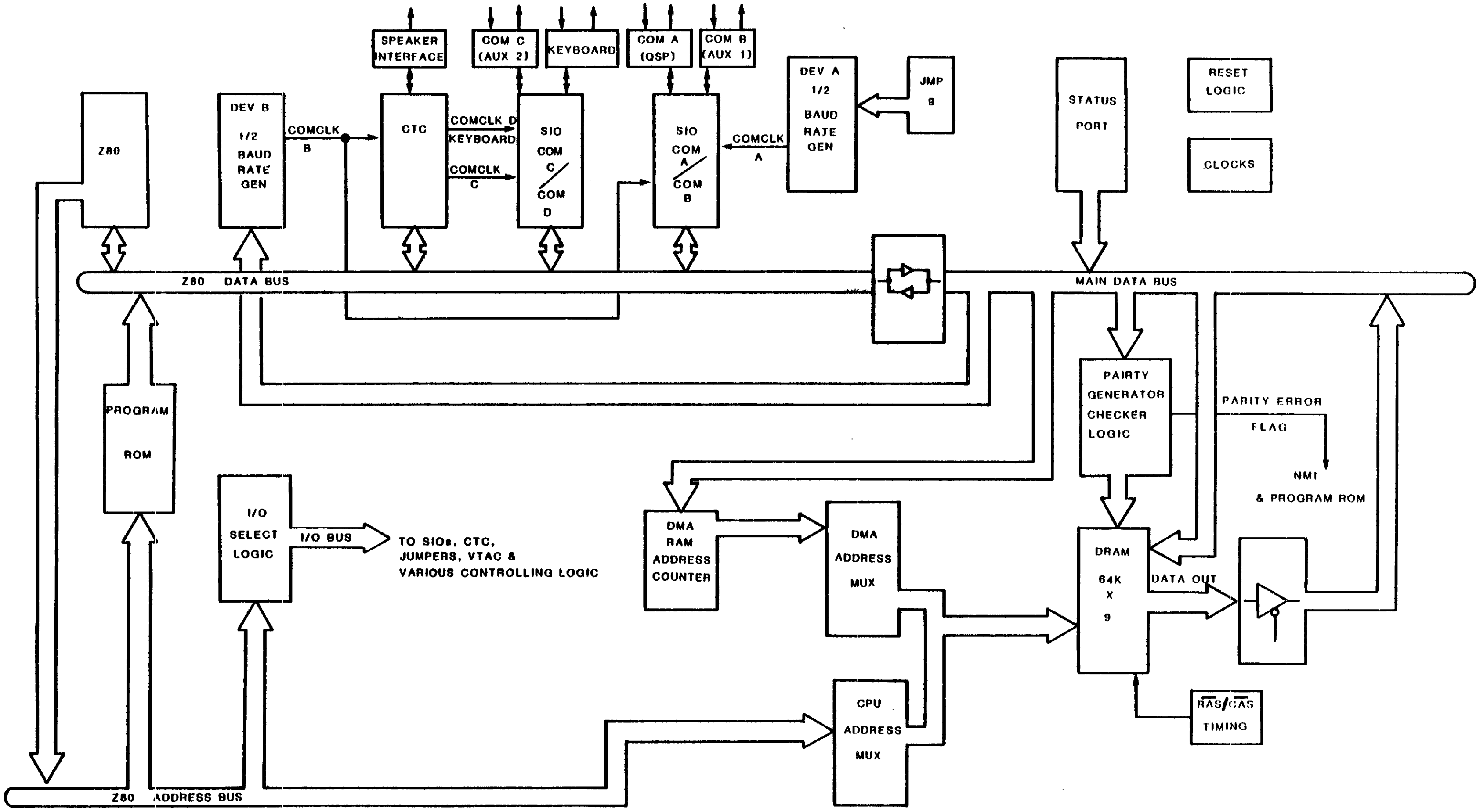
CLEAR Same as in non-right-justified mode.

SHIFT/CLEAR Same as in non-right-justified mode.

TAB Same as in non-right-justified mode.

SHIFT/TAB Same as in non-right-justified mode.

TRANSMIT Same as in non-right-justified mode.



VT4R CPU BLOCK DIAGRAM

Figure 6-1

EXPLANATION OF CPU LOGICS

Refer to the CPU Block Diagram (Figure 6-1) and the CPU Logic Drawings for the following:

When the VT4 is first powered on, a $\overline{\text{RESET}}$ signal is generated. This forces the Z80's Interrupt Register, the I and R Registers, and the Program Counter to be cleared. When $\overline{\text{RESET}}$ goes high, the Z80 begins executing the initialization program in the Program ROM.

The VT4 will first power up emulating a VT3. The microcode initializes the CTC, the two SIOs, loads the keyboard tables, and reads the jumpers for the QSP address, QSP baud rate, and 50/60 Hz setting. The VTAC on the Video Board is also initialized at this time.

The VTAC, CTC, and the SIOs contain their own internal registers and they can generate their own interrupt routines. The CTC and the SIOs each have an Interrupt Enable In (IEI) and an Interrupt Enable Out (IEO). When one of these chips pulls $\overline{\text{INT}}$ low, it also drops IEO. This in turn inhibits any subordinate chip from generating an interrupt. The CTC has the highest interrupt priority, the SIO handling COM A and COM B has the next level of priority, and the SIO handling COM C and COM D has the lowest priority. Furthermore, the CTC and SIOs have internal levels of interrupt priorities. The highest level interrupt is generated by the CTC when it receives $\overline{\text{FLN}}$ from the Video Board when it is time to flag the Z80 to update the DMA Counters. Each SIO is able to keep track of its own communications channels.

The Dynamic RAM stores instructions and data used by the Z80 for VT3 emulation and downloaded user code. It also stores data for the screen display, interrupt vectors, and QSP buffers.

Screen display data is accessed directly by the Video Board after the Video Board has generated an interrupt. The interrupt routine initializes the DMA counter and then generates $\overline{\text{BUSREQ}}$. The Z80 answers with $\overline{\text{BUSAKZ80}}$ at the start of the next machine cycle. This forces the Address Bus, the Data Bus, $\overline{\text{IOREQ}}$,$\overline{\text{MEMREQ}}$, RD, and WR to go into a high impedance state so the DMA counters (on the CPU Board) can control these lines for a DMA.

When $\overline{\text{BUSAKZ80}}$ is active, it will disable the CPU Address Multiplexer and BUSAKZ80HI will enable the DMA Multiplexer. Buffer 1 on the Video Board is loaded at this time. When the buffer is filled, the Video Board issues $\overline{\text{BUSREL}}$. This sets $\overline{\text{BUSREQZ80}}$, which enables the CPU Address Multiplexer. At the same time, BUSAKZ80HI goes low and disables the DMA Address Multiplexer.

The parity bit always starts out high on a write to memory. Parity is checked during a subsequent read from memory. Parity is established on the write when the Even Parity Output of the Parity Generator/Checker is brought around and inverted. When the other eight bits are read back with the new parity bit, parity will be odd (unless, of course, one of the eight data bits gets turned over between the write and the read). If a parity error occurs, $\overline{NMI}$ will be pulled low, the program will branch to 0066H, the Parity Flag will be checked, and the program will go into a halt loop.

Page 2 CONNECTORS ON THE CPU BOARD

This page shows the interconnections between the CPU Board, the Video Board, and the outside world.

J1. This is the interface to the Video Board. It has all the Z80 Address Bus and Data Bus lines. Timing signals are also interchanged between the CPU Board and the Video Board, using J1.

J2. This is the Power Supply Connector. There is a parallel J2 connector on the Video Board. Both J2s are connected to the power supply by a common cable.

J4. This connector at the CPU Board's bottom (it appears up because the board is mounted upside-down) is reserved for the Z80 Test Panel.

J5. This is Main A, or the QSP port to and from the IOU 39Q.

J6. This is Main B, or the QSP port to and from the next terminal on the QSP string.

J7. This is AUX 1.

J8. This is AUX 2.

J9. This connects to the contrast, brightness, and volume control potentiometers; and it also interfaces to the keyboard.

JMP1 This connects +12v on J6, pin 20 from J5, pin 20.

JMP2 This connects -12v on J6, pin 22 from J5, pin 22.

EXPLANATION OF CPU LOGICS

Page 3 Z80 MICROPROCESSOR WITH DATA AND ADDRESS BUSES

This page shows the Z80 and all the signals transmitted and received by it.

The chip located at 6G is a bi-directional, three-state buffer. When the buffer is enabled, the Z80 has access to the Data Bus. When the buffer is disabled, the Data Bus is isolated from the Z80 and is available for DMA.

The signal $\overline{\text{RDZ80}}$ controls the direction of the data on the bus when 6G is enabled. When $\overline{\text{RDZ80}}$ is low, the Z80 can read the Data Bus. When $\overline{\text{RDZ80}}$ is high, the data bus gets loaded with what is on the Z80 data lines at the time.

PARERRHI is a parity error signal that will pull a non-maskable interrupt and restart the Z80 from address 0066H.

The flip-flop at location 3E will be set whenever there is a Power-On-Reset Signal to the Z80, or whenever a DMA device releases the Data Bus. $\overline{\text{SELIO58}}$ (DMA count and start DMA) will generate a $\overline{\text{BUSREQ}}$ Signal to the Z80. This status will be held by the flip-flop until the requesting device sends back a $\overline{\text{BUSREL}}$.

EXPLANATION OF CPU LOGICS

Page 4 Z80 ADDRESS MULTIPLIXER AND POWER ON RESET CIRCUITS

Z80 Address Multiplexer

Two 8-bit to 4-bit multiplexers are used to shift out the low byte and then the high byte of RAM address as requested by the Z80.

Low bits 0-3 are connected to the A inputs 0-3 respectively on the multiplexer at 5D, and low bits 4-7 are connected to the A inputs 0-3 respectively on the multiplexer at 5C. Similarly, the high bits 8-11 are connected to B inputs 0-3 respectively on chip 5D, and high bits 12-15 go to B inputs 0-3 respectively on chip 5C.

When SADDRMUX (Strobe Address Multiplexer Signal) is low, the low byte of the RAM address will go out onto the RAM Address Bus as soon as $\overline{\text{ADDRDIS}}$ and $\overline{\text{BUSAKZ80}}$ are both inactive (high). When SADDRMUX is high, the high byte of the RAM address will go out onto the RAM Address Bus as soon as $\overline{\text{ADDRDIS}}$ and $\overline{\text{BUSAKZ80}}$ are both inactive. Note that transitions of SADDRMUX occur when the multiplexer chips are disabled.

Power-On-Reset Circuit

The circuit at the bottom of the drawing is the Power-On-Reset Circuit. When first powered on, Capacitor C2 is fully discharged, causing the voltage at Pin 3 of the comparator chip (LM311) at 12A to be low. Voltage at Pin 2 comparator will be about 2.5v, which is high compared to Pin 3. The LM311 will output a TTL level high voltage which is inverted. This is $\overline{\text{RESET}}$. A signal coming off of $\overline{\text{RESET}}$ is inverted (RESETHI) and goes through another inverter to $\overline{\text{RESET}}$ on the Z80. The original $\overline{\text{RESET}}$ Signal goes to reset the other logic in the terminal.

When Capacitor C2 charges up sufficiently (after approximately 68ms) voltage at Pin 3 on the LM311 will go up to +5v and the output will go low. All resetting signals will then go inactive.

Page 5 DMA ADDRESS MULTIPLEXER

The counter chips (74191s) are initialized in the following manner: The low byte of data is first loaded onto the Data Bus. The low byte will be loaded onto chips 11C and 3C when $\overline{\text{SELIO68}}$ becomes active (low). The high byte of data is then loaded onto the Data Bus and is followed by $\overline{\text{SELIO70}}$ which clocks the high byte into chips 8C and 6C.

The multiplexer chips shown on this page are 74257s as on page 4. These Chips are enabled when BUSAKZ80HI is high (the Z80 has acknowledged the bus) and $\overline{\text{ADDRDIS}}$ is inactive (high). $\overline{\text{VMUX}}$ will go low to select the low byte of the current DMA address. Then the multiplexers will become enabled and allow the low byte of the current DMA address to be strobed onto the 8-bit RAM Address Bus. $\overline{\text{VMUX}}$ will go high to select the high byte of the current DMA address. Then the multiplexers will become enabled and allow the high byte of the current DMA address to be strobed onto the 8-bit RAM Address Bus.

Each 74LS191 counter has a ripple carry output that enables the next sequential counter for one clock pulse. Since the counters are all clocked by $\overline{\text{VCAS}}$, each counter will change synchronously with the upward transition of $\overline{\text{VCAS}}$ as long as its enable input is low at the time of the occurrence of the upward edge of the $\overline{\text{VCAS}}$ pulse.

EXPLANATION OF CPU LOGICS

Page 6 DRAM

The Dynamic RAM is arranged using nine chips of 64K by 1-bit. The combined outputs of eight chips at any one time constitute one byte. The ninth chip (1B) is for parity checking.

Each chip has one line from the Data Bus connected to its input and another line from its output to the RAMDOUT Bus. $\overline{\text{RAMWRT}}$ is connected to the active low WRT pin of each RAM chip. Write is enabled when $\overline{\text{RAMWRT}}$ is low, and read is enabled when $\overline{\text{RAMWRT}}$ is high.

When the memory is addressed, the low byte of address is first placed onto the Address Bus. Then $\overline{\text{RAS}}$ (Row Address Strobe) is set low while $\overline{\text{CAS}}$ (Column Address Strobe) is held high. Next, the high byte of address is placed onto the Address Bus while $\overline{\text{RAS}}$ remains low. Then $\overline{\text{CAS}}$ is set low. This completes the read or write, depending upon the state of $\overline{\text{RAMWRT}}$.

Any time the memory is addressed, a refresh is automatically accomplished. Since memory must continually be refreshed whether or not it is being accessed, refresh cycles are continuously issued to the memory.

A refresh only cycle is accomplished by sequencing through each Row Address and the clocking $\overline{\text{RAS}}$ low. The Row Address is incremented while $\overline{\text{RAS}}$ is high. $\overline{\text{CAS}}$ is always held high during a Refresh Only Cycle. All memory cells in an entire row are refreshed when $\overline{\text{RAS}}$ goes low. As long as $\overline{\text{CAS}}$ is high, $\overline{\text{WRT}}$ and DIN don't care.

Page 7 PARITY GENERATOR/CHECKER

Parity is generated after a write to the memory. Parity is checked whenever memory is read. The Parity checking is accomplished by means of the following sequence of events:

Data to be written to memory is loaded onto the Data Bus. $\overline{\text{MEMRD}}$ goes high to enable a write. $\overline{\text{BUSKZ80}}$ will also be high during a write. This generates a signal equivalent to $\overline{\text{MEMWR}}$ on Pin 10 of Chip 5C which forces a logical 1 on Pin 4 of the parity checker Chip 12B. If the number of logical 1s on the inputs of 12B (including the parity bit) is odd, PARIN will be zero. If the number of logical 1s on the inputs 12B is even, PARIN will be 1. Pin 6 will be the opposite of Pin 5. Odd parity is thereby generated during the write.

When $\overline{\text{MEMRD}}$ becomes active (low), Pin 10 on Chip 5C will go high and the RAMDOUT Bus going into the three-state buffer (3A) will go out onto the Data Bus. PARIN (the bit at Pin 5 on 12B), which was written to the Parity RAM Chip (1B), comes back to Chip 5C, Pin 9 as PAROUT. This bit is inverted by 5C. The resulting parity will be odd, unless one of the bits on the Data Bus has changed between the time it was written to the RAM and the time it was read.

The PARCLOCK Signal will clock the Parity Odd bit (Pin 6 on 12B) through the flip-flop at 10E. As long as the Parity Odd Output (Pin 6 12B) is high during PARCLOCK, there is no parity error.

EXPLANATION OF CPU LOGICS

Page 8 MEMORY ACCESS AND DOT TIMING

This sheet illustrates how the various control signals are generated by the CPU Board.

The "heart" of the entire board is the 48MHz hybrid oscillator at 11H. The 48MHz pulse gets buffered through one of the OR Gates (which acts as an amplifier and squares the pulse) in Chip 12H. Then the pulse is cascaded through three flip-flops in the 74F161 counter chip located at 10H. The signals derived are 24MHz, 12MHz, and 6MHz clocks.

The 48MHz Signal is also divided by three at Chip 11G to generate DOT16 (16MHz). This is an alternate dotclocking signal used to widen the displayed characters when the terminal is in 64 character mode (the 24 MHz signal is inverted to become DOTCLOCK for 80 character mode).

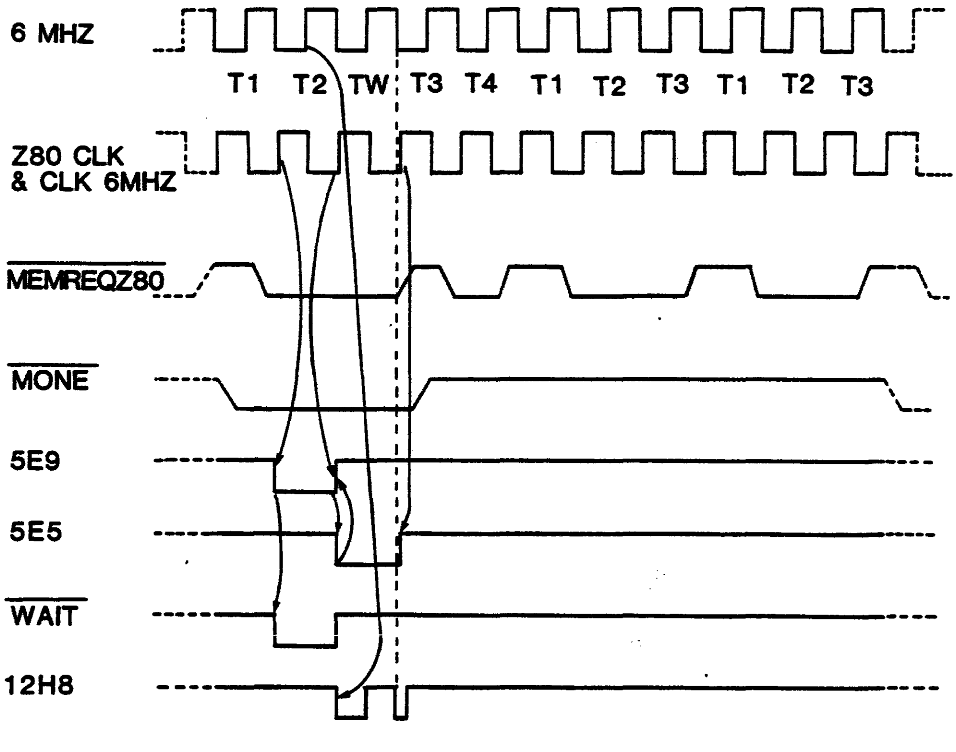


Figure 6-2

Timing Diagram in Figure 6-2

When MONE is low, the next upward clock transition will reset Pin 9 of Dual Flip-Flip 5E. If MEMREQZ80 is also low (as it normally will be during most of an MONE cycle) WAIT- will go low. This will occur during T2 and will force an automatic Tw state between T2 and T3, because the low signal at 5E9 will also force a low to appear on the following upward clock pulse on Flip-Flop 5E5. The low on 5E5 will set 5E9 and WAIT will go high. This will be the start of the above-mentioned Tw state. 5E5's forcing 5E9 high will finally be fed back into 5E5 on the upclock at the end of Tw. When 5E5 goes high, T3 will start.

12H8 will go low whenever 5E5 and 6MHz (compliment of CLK6MHz) are low.

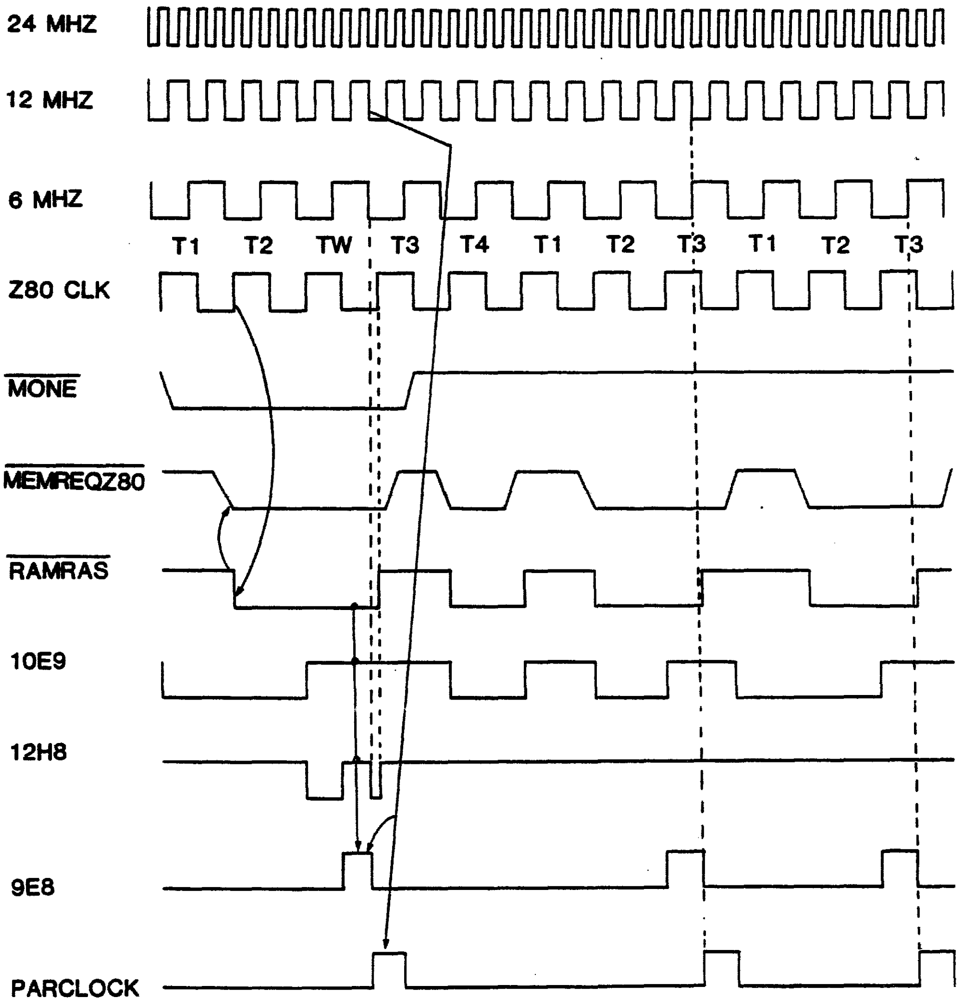


Figure 6-3

Timing Diagram in Figure 6-3

When MEMREQZ80 goes low, the next upward transition of CLK6MHz (the Z80 Clock) will drive RAMRAS low. When RAMRAS is low (the compliment signal-- RAMRAS going high-- is actually what is used) and 10E9 and 12H8 are high, 9E8 will go high. A high on 9E8 and an up pulse from the inverted 12MHz Signal will drive PARCLOCK high. The compliment of PARCLOCK is fed back to set RAMRAS. RAMRAS will remain high until MEMREQZ80 goes low and an upward transition of the Z80 Clock occurs.

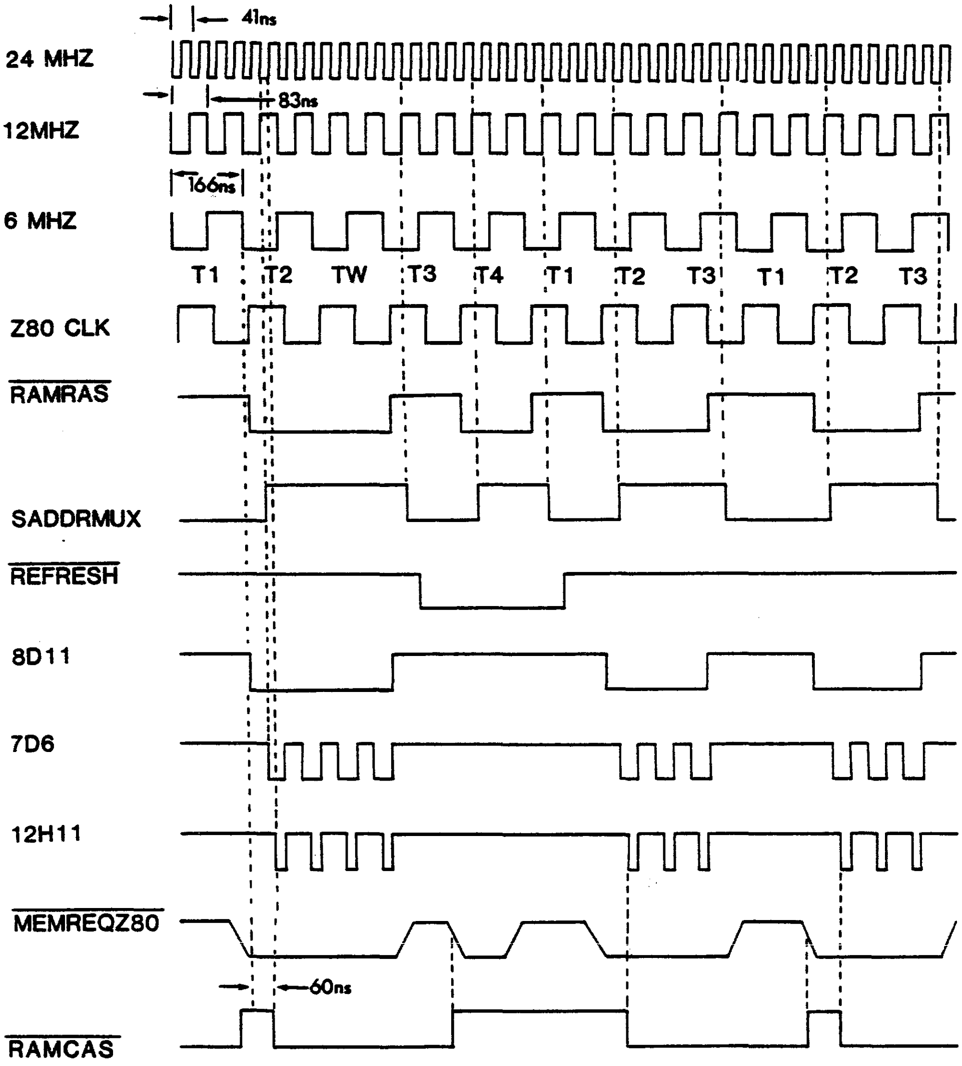


Figure 6-4

Timing Diagram in Figure 6-4

SADDRMUX goes high when the compliment of $\overline{\text{RAMRAS}}$ goes high and the upward edge of DOTCLOCK (compliment of 24MHz on Figure 6-4) is encountered. When $\overline{\text{REFRESH}}$ is high and $\overline{\text{RAMRAS}}$ is low, 8D11 goes low. 7D6 goes low when 8D11 is low and the compliment of 12MHz (after being inverted by 9G5) goes low. 12H11 lags 7D6 by approximately 20ns, because 12H11 can only go low when DOTCLOCK (which is the compliment of 24MHz) goes low.

$\overline{\text{RAMCAS}}$ is generated at the $\overline{Q}$ output of a J-K flip-flop in Chip 9D. The downward transition of $\overline{\text{MREQZ80}}$ forces $\overline{\text{RAMCAS}}$ high. A downward pulse from 12H11 sets the flip-flop and causes $\overline{\text{RAMCAS}}$ to go low. Since $\overline{\text{MREQZ80}}$ acts as a clock to set $\overline{\text{RAMCAS}}$, and the Z80 clock's upward transition is what resets $\overline{\text{RAMRAS}}$, $\overline{\text{RAMCAS}}$ will be high during $\overline{\text{RAMRAS}}$'s downward transition. $\overline{\text{RAMCAS}}$ will go low approximately 60ns after $\overline{\text{RAMRAS}}$.

BUSAKZ80HI is used as a selector for Multiplexer 11A. When BUSAKZ80HI is high, it means that the Z80 has relinquished the Data and Address Buses and that the Video Board can use DMA. The multiplexer will pass the $\overline{\text{VRAS}}$ and $\overline{\text{VCAS}}$ Signals from the Video Board to the $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ lines to the RAM. When BUSAKZ80HI is low, the Z80 is using the buses and $\overline{\text{RAMRAS}}$ and $\overline{\text{RAMCAS}}$ Signals will be passed through the multiplexer to the $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ lines to the RAM. $\overline{\text{RAS}}$ selects the eight bits and $\overline{\text{CAS}}$ selects the upper eight bits of the chosen memory location.

Cursor and Character Blink

Chip 3H is a dual, four-bit binary counter. Both counters are cascaded together to make the chip function as an eight-bit binary counter. This counter divides VDRIVE (the vertical drive signal from the Video Board) by 256 to generate a 4.2Hz signal called BLINKCLK. BLINKCLK blinks any characters that have the Blink Attribute turned on.

EXPLANATION OF CPU LOGICS

Page 9 ROMs AND ROM SELECT

The flip-flop at 5D controls whether or not the ROMs are selected. It does this in one of two ways: 1) A Power-On-Reset or a parity error resets the flip-flop, thus generating $\overline{\text{ROMMODE}}$. 2) Data Bit 7 is low when $\overline{\text{SELI038}}$ goes from low to high. When $\overline{\text{ROMMODE}}$ is high, RAM is selected. The first 3000H bytes of RAM and ROM overlap when standard VT4 ROMs are used.

Whenever a ROM is selected, one of the three ROMSEL Signals is ORed (negative logic) to obtain $\overline{\text{ROMEN}}$ at Pin 6 of Chip 9E. When $\overline{\text{ROMEN}}$ is low and $\overline{\text{MEMRD}}$ is low, the octal bus transceiver chip 6G is disabled ($\overline{\text{MEMRD}}$ and Chip 6G are shown on Page 3 of the CPU Logic Drawings). $\overline{\text{ROMEN}}$ also disables the three-state buffer at 3A (Page 7 of the CPU Logics). This insures that RAM cannot be accessed while ROM is accessed.

$\overline{\text{SELI050}}$ enables four bits of a three-state buffer at 10C. This allows the state of $\overline{\text{ROMMODE}}$ to be preserved in case of an interrupt. After a return from interrupt, $\overline{\text{ROMMODE}}$ will be low if a ROM routine was interrupted and $\overline{\text{ROMMODE}}$ will be high if a RAM routine was interrupted.

2732 ROMs

When ROM type 2732s (standard VT4 ROMs) are used, ROM addresses will be from 0000H through 2FFFH.

When 2732s are used, JMP 5 must be set with pin patches going from Pins 1 to 2, 3 to 4, and 5 to 6. This allows Address Lines 12 and 13-- through the multiplexer at 10D-- to select the current ROM chip. Address line 14 will always be low when 2732s are used and JMP 5 is properly set. Address line 15 will always be low when ROMs are enabled.

ROMs are selected as follows:

ADDR13	ADDR12	Chip No.
0	0	0
0	1	1
1	0	2

Caution: The ROM sockets will accept 28-pin 2764 ROMs; however, when 24-pin 2732s are used, each ROM must be installed so that Pin 1 of the ROM chip plugs into Hole 3 of the ROM socket.

2764 ROMs

When ROM type 2764s are used, ROM addresses will be from 0000H through 4FFFH.

When 2764s are used, JMP 5 must be set with pin patches going from Pins 2 to 3, 4 to 5, and 6 to 7. This allows Address Lines 13 and 14-- through the multiplexer at 10D-- to select the current ROM chip. Since one more address line is used to select a 2764 ROM chip than is used to select a 2732 ROM chip, available ROM is doubled. Address line 15 will always be low when ROMs are enabled.

ROMs are selected as follows:

ADDR14	ADDR13	Chip No.
0	0	0
0	1	1
1	0	2

EXPLANATION OF CPU LOGICS

Pages 10 and 11 SIOs, CTC, QSP PORT, AND AUX 1 & AUX 2 PORTS

The Z80B, the Z80 clock, and the COM 8116 Dual Baud-Rate Generator are all used to control the data rates of one or more of the four SIO channels.

The COM 8116 uses two separate channels to generate two baud rates from one 4.9 MHz crystal.

The 8116's Channel T is used to generate the baud rate for Port A, the QSP port. This channel's baud rate is selected by the four pin patch jumpers on JMP9 closest to the edge of the board (MSB is closest to the edge of the board). The jumpers pull the lines low when they are installed. Since a logical binary 1111 generates a 19.2K baud rate, all jumpers should be removed for 19.2K baud (This is normal data transmission speed for a QSP line connected directly or through line drivers to the computer). The chart in the appendix shows all the possible baud rate settings for this chip; however, caution should be exercised since many of these settings are not compatible with the IOU 39Q. The baud rate settings shown in Section 2.1 are all IOU 39Q compatible.

Channel R of the 8116 generates a timebase that is controlled by the Data Bus. The output of Channel R, COMMCLKB, clocks TxCB and RxCB on the SIO at 4F (on Page 11). This goes to I/O Port B on AUX 1. COMMCLKB also goes to Channel 2 of the Z80B-CTC. The CTC is capable of dividing COMMCLKB down, but the Power-On sequence programs the CTC to pass the COMMCLKB Signal without dividing it. Therefore XMITB (AUX 1) and XMITC (AUX 2) are the same frequency when the VT4 is powered on.

CTC Channels 0 and 1 are in Timer Mode. Channel 1 has the Z80 Clock input divided by 16 (using the 16/256 prescaling option for the Z80 CTC) when the VT4 is powered on. It is possible for downloaded microcode to change the prescaler to 256 instead of 16.

The CTC is programmed to generate an interrupt on Channel 3 after a count of one. $\overline{\text{FLN}}$ is fed into the input of Channel 3 to generate an interrupt for the screen refresh routine.

The VT4 uses two Z80B SIO-2s. The SIO on Page 10 (Chip 2F) interfaces the Parallel Data Bus with the Serial Keyboard and with the serial I/O Port AUX 2. The SIO on Page 11 (Chip 4F) interfaces the Parallel Data Bus with the Serial QSP Port A and with the Serial I/O Port AUX 1.

Of the three devices discussed, the CTC has the highest interrupt priority. The SIO at 4F has the next level interrupt priority, and the SIO at 2F has the lowest interrupt priority.

The CTC and the two SIOs each have an IEI (Interrupt Enable In) and an IEO (Interrupt Enable Out). Any time an IEI goes low, IEO on the same chip will go low. The lines are daisychained from the CTC to the SIO at 4F to the SIO at 2F. The CTC's IEI line is held high at all times. During an interrupt, the CTC's IEO line is driven low by the chip's own internal logic. This daisychain insures that an SIO cannot generate an interrupt while a higher priority device has an interrupt pending.

EXPLANATION OF CPU LOGICS

Page 12 I/O DECODER

The two 74LS138 Demultiplexers decode Address Lines 3-7 into I/O Ports 08H-F8H. Address Bits 0-2 can be used as a device control when they are included as part of the I/O Port Address.

During the execution of an I/O instruction, Chip 3G is selected when Address 6 (40H-weight bit) is off and Chip 7G is selected when Address 6 is on.

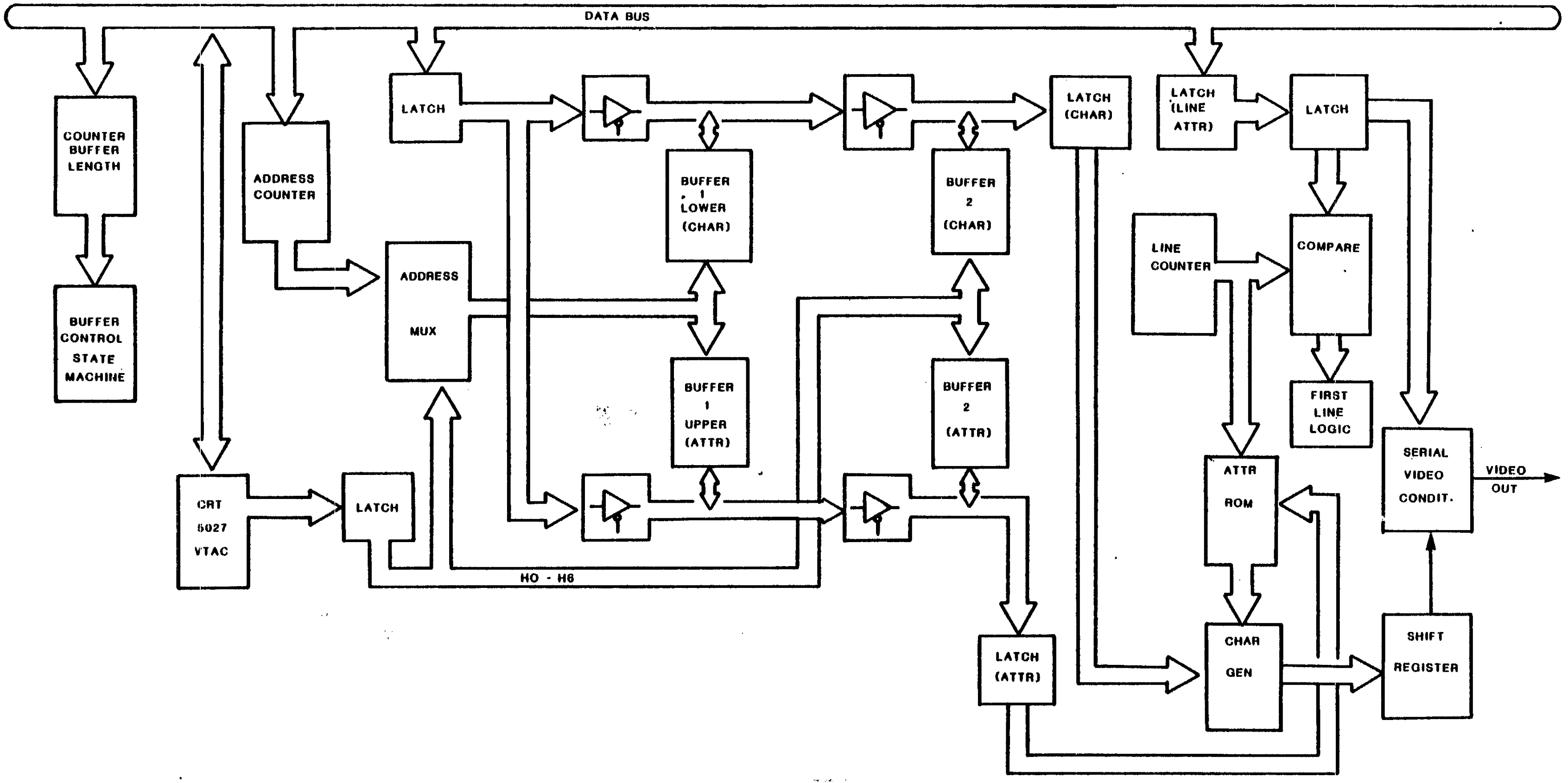
When Address 7 (80H-weight bit) is turned on during an I/O instruction, the VTAC is always addressed. $\overline{\text{WRTZ80}}$ and $\overline{\text{IOREQ}}$ must be low and Address 7 must be high in order to allow $\overline{\text{VTACSTB}}$ to toggle and strobe the VTAC. All other conditions will cause $\overline{\text{VTACSTB}}$ to be stuck high.

If a 1 is written to Data Bit 0 during an output to Port 78, the flip-flop at 11D will enable the 64 character clock (16MHz). If a 0 is written to Data Bit 0 during an output to Port 78, the 80 character clock (24MHz) will be enabled (see Video Logics, Page 4). During the Power-On Sequence, the microcode sets up 64 Character Mode by reading Test 1 on JMP 7 (shown on page 13), provided the jumper is installed. The Power-On Sequence eventually sets Data Bit 0 high.

Page 13 JUMPER DECODERS

After the Power-On/Reset Sequence, the ROM initialization routine takes over. During ROM initialization, the program selects one of the three-state buffers and reads the jumpers. The program uses the data from the jumpers and then selects another three-state buffer, whose jumpers are again read and whose data is again processed. Only one three-state buffer is enabled at any time, thereby eliminating conflicting data from the Data Bus.

Chips 9C and 10B are cascaded 4-bit counters used to count characters being written to the Display RAM. Pin 5 is held high on both of these chips to cause them to count down. When I/O 58 is selected, the number of bytes of data is loaded into the counters. As R/$\overline{\text{W}}$BUF1L is toggled, data is transferred and the counters are decremented by one. When the counters reach zero, $\overline{\text{FIN}}$ is generated.



VT4R VIDEO BLOCK DIAGRAM

Figure 7-1

EXPLANATION OF VIDEO LOGICS

There are two sets of display buffers: Buffer 1 and Buffer 2. First, Buffer 1 gets loaded with Character and Attribute data during a DMA read. When the Buffer 1 load sequence finishes, data is transferred to Buffer 2. A 6309 PROM is used to control the sequencing of the display buffer reads and writes.

The main CPU Memory originally stores the screen data in a character - attribute - character - attribute fashion. When the Display Logic requests data for the next character row, an interrupt request is sent to Channel 3 of the Z80 CTC (counter-timer circuit). The CTC will in turn issue an interrupt to the Z80B. There will be up to a 2.7 us delay from the time $\overline{\text{FLN}}$ goes low to the time that the interrupt line is actually pulled down.

An interrupt starts the following:

1. A DMA Read of CPU Memory starts. The Data Bus receives display data. First a character is read and then written to BUF1L (the lower half of Buffer 1), and then an attribute is read and then written to BUF1U (the upper half of Buffer 1). The Character Counter (which has been pre loaded with the character count for the row) is decremented.

2. While Buffer 1 is being filled, Buffer 2 is being read. The first scan line of all chracters in Buffer 2 starts to be displayed as soon as $\overline{\text{FLN}}$ goes low.

3. After Buffer 1 has been filled and Buffer 2 has been displayed, new data in Buffer 1 is transferred to Buffer 2. In this manner, screen data is continuously pipelined to the display.

The actual display of characters is done from Buffer 2. Each character time, two bytes of character data (character and attribute) are fetched from Buffer 2.

The characters in the Character Generator are arranged in horizontal character slices. The Character Generator uses 12 address bits to define eight output bits. The lowest four bits define are the row address of the Character Generator, and they remain the same during the display of an entire scan line. The upper eight bits correspond to the hex code of the ASCII character being displayed. The upper eight bits will change each time a new character column is about to be displayed on a scan line. The upper eight bits come from Buffer 2. Therefore, Buffer 2's output will cycle through all the characters on a data row times the number of scan lines per data row. This would typically be 80 X 12.

After the last scan line of a character row is completed, Buffer 2 must be updated. Meanwhile, Buffer 1 has been getting loaded with the next character row of data during the last character row's display cycle.

When $\overline{\text{FLN}}$ goes low, that is the signal for the CTC to request an interrupt. A transfer will occur before the next DMA Cycle starts. While characters are being transferred to Buffer 2, the first scan line will start displaying because Buffer 2's I/O Bus is connected through a latch to the Character Generator and display circuitry.

Page 2 INTERCONNECTIONS

This page shows the interconnections between the Video Board, the CPU Board, the Monitor, and the Power Supply.

J1. This is the interface to the CPU Board. It has all the Z80 Address Bus and Data Bus lines. Timing signals are also interchanged between the CPU Board and the Video Board using J1.

J2. This is the Power Supply Connector. There is a parallel J2 connector on the CPU Board. Both J2s connect to the Power Supply by a common cable.

J3. This is the interface to the Monitor. All video output signals and sychronization signals use this connector.

Page 3 BUFFER CONTROL STATE MACHINE

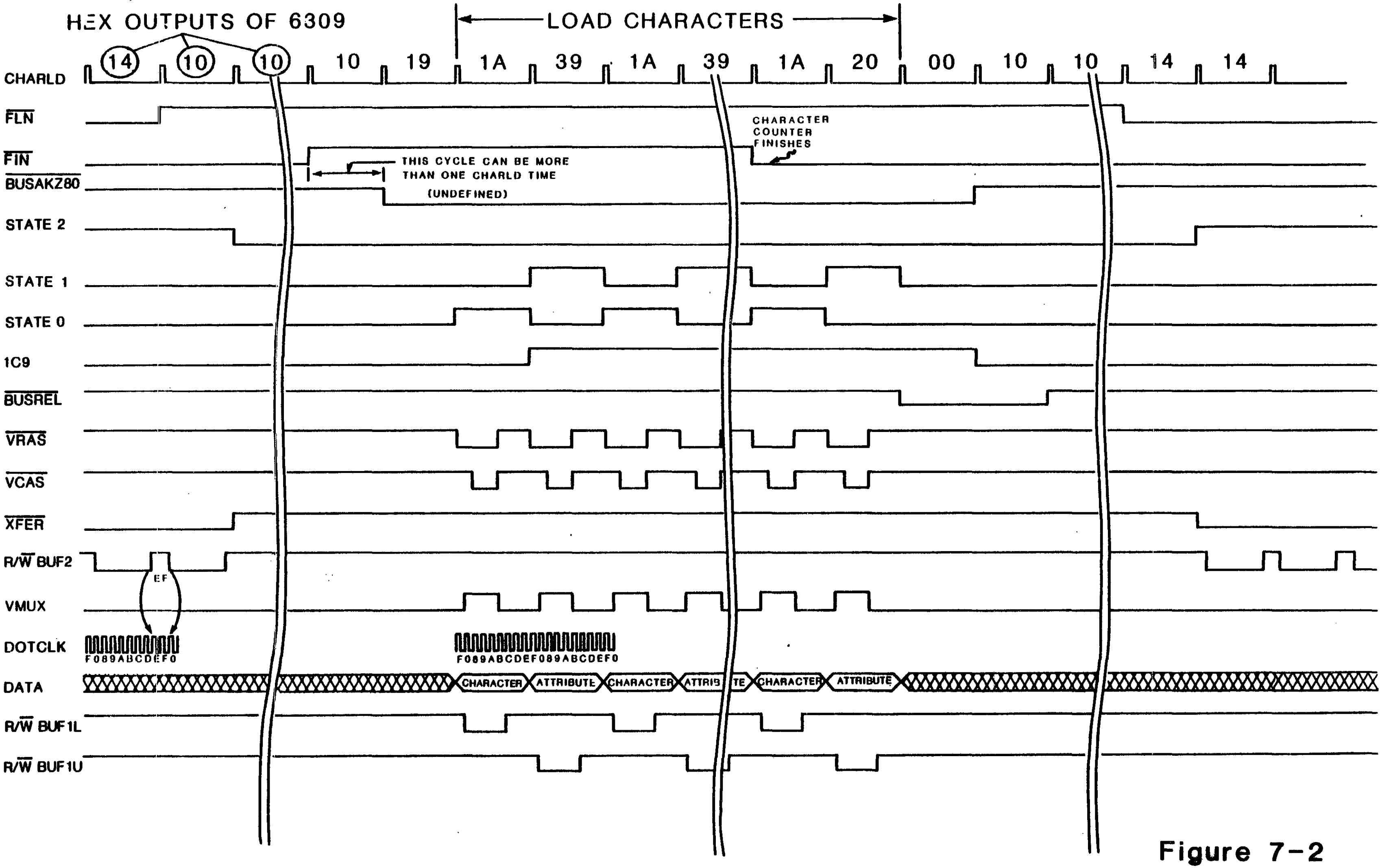


Figure 7-2

EXPLANATION OF VIDEO LOGICS

A 6309 PROM and an 8 X D flip-flop chip for a latch are used to control the DMA transfers from Main Memory to display buffers and pipelining from Buffer 1 to Buffer 2. The flip-flop chip at 1C feeds back to the 6309 so that current outputs and controls ($\overline{\mathrm{FLN}}$, $\overline{\mathrm{FIN}}$, and $\overline{\mathrm{BUSAKZ80}}$) point to the next state and determine the next output.

The timing diagram Figure 7-2 shows the various phases of the signals and how they interact. The numbers across the top of the timing diagram are the hexadecimal outputs of the 6309.

The ROM's address pins, A6,A5,A4,A2,A1,A0 respectively are selected by $\overline{\mathrm{FLN}}$, $\overline{\mathrm{FIN}}$, $\overline{\mathrm{BUSAKZ80}}$, State 2, State 1, and State 0. The chip address pins A7 and A3 are wired low, allowing only one fourth of the chip's capacity to be used. The states of the 6309's address lines determines the chip's hex output.

When a new character row is about to be loaded into Buffer 1, $\overline{\mathrm{BUSAKZ80}}$ will be low. The output of the 6309 will be 19H, which signals the buffer control logic that a DMA is about to begin. The next CHARLD will generate a 1AH on the 6309 and the first character in the data row will be loaded into the lower half of Buffer 1. The following CHARLD will generate a 39 and the first character's attribute will be loaded into the upper half of Buffer 1. The character counter (shown on the bottom of page 13 of the CPU logics) will decrement the next time R/$\overline{\mathrm{W}}$BUF1L goes low. Note that loading the character counter starts the DMA.

When the character counter finishes, $\overline{\mathrm{FIN}}$ will go low. The next CHARLD will cause the 6309 to generate its final 1AH and the final character in the row will be loaded into the lower half of Buffer 1. The last attribute will be loaded into the upper half of Buffer 1 at this time.

The next CHARLD causes the 6309 to generate a 00H and causes $\overline{\mathrm{BUSREL}}$ to go low, returning the Data Bus to the Z80. The next CHARLD begins a series of generations of hex 10s on the 6309's outputs. This gives the display logic the extra time it needs to finish displaying the contents of Buffer 2, slice by slice, on the screen.

When $\overline{\mathrm{FLN}}$ goes low and the previous output of the 6309 was 10H, the next CHARLD will cause the 6309's output to change to 14H. The outputs will continue to be 14H during subsequent CHARLDs while data is being transferred from Buffer 1 to Buffer 2. During the transfer, the first scanline of Buffer 2 characters will be displayed as Buffer 2 is being loaded.

After the character counter for Buffer 2 (located on the 5027 VTAC chip shown on Page 4) is empty, $\overline{\mathrm{FLN}}$ will go high. The next CHARLD will cause the 6309 oututs to be 10H, and the last character in the line (and its attribute) will be written to

Buffer 2. When $\overline{\text{FIN}}$ goes high, the next DMA cycle will be ready to begin.

The signal, DOTCLK at drawing coordinates B5 is derived from DOTCLOCK when the terminal is in the 80/96 Character Mode, and from DOT16 when the terminal is in 64 Character Mode. This insures that the ratio of DOTCLK pulses to CHARLD pulses is always nine to one regardless of the terminal's display mode, because CHARLD is derived from DOTCLK. This will be discussed on the next page of the Video Logics.

DOTCLK and CHARLD are shown on this page to generate $\overline{\text{VRAS}}$, then VMUX, and then $\overline{\text{VCAS}}$ each on a consecutive DOTCLK pulse. Timing diagram Figure 7-2 shows the sequencing of all the output signals on this page.

Page 4 VIDEO TIMING CONTROL

The two dot clocking signals used in the circuits shown on this page come from the CPU Board (Page 8 of the CPU Logics). These two signals arrive at the multiplexer chip at 1D on the Video Board (upper left of this page). When the control signal EN64CH is low, the multiplexer selects DOTCLOCK (24 MHz). When EN64CH is high, the multiplexer selects DOT16 (16 MHz). The resultant signal is output by the multiplexer and shown as DOTCLK. DOTCLK is the timebase for all the Video Logic: 16 MHz is for 64 Character Mode and 24 MHz is for 80/96 Character Mode.

DOTCLK is next used to clock a 4-bit, binary, sychronous counter located at 1G. The counter is hardwire programmed to count up from 8H through FH. After the FH count a 0H is forced onto the counter's outputs and a logical 1 is forced onto the ripple carry output of the counter. The rollover after the FH count forces all the other outputs of the counter to 0. The 0 is fed back to the load enable input of the counter, causing the count to restart from 8H.

The signal that pulses from the ripple carry output of counter 1G is CHARLD, which has already been discussed on page 3 of the Video Logics. CHARLD is also used to clock the counters in the 5027 VTAC at 2J, and to clock the VTAC's output flip-flops 3E and 1H.

The VTAC is initialized by the VT4 Microcode when 64 or 80/96 Character Mode is selected. The VTAC is used in the VT4 to control scan line count, character count, blanking, horizontal and vertical synch pulses, and scan lines per frame (used to synch 50 or 60 Hz).

Character and Scan Line Counters

The VTAC's outputs H0 - H6 are the internal character counter outputs. These outputs are connected to the octal flip-flop at 3E. CHARLD increments the character counter and latches the counter's previous output levels on the flip-flop. (This is possible due to the internal propagation delay of the VTAC.)

The scan line count is incremented at the end of the character count, and (if it isn't the last scan line of the character row) the character count is reset. The character count then increments again for length of the next scan line.

Blanking

Blanking is used to turn off the CRT's electron beam when it is scanning beyond the set boundaries of the video display and during a retrace.

Horizontal Synchronization

HSYNC is generated by the VTAC to sychronize the horizontal sweep. This signal goes to a one-shot in Chip 1K. The one-shot stretches the pulse and generates HDRIVE. HDRIVE goes to the Monitor Board to control the horizontal sweep of the CRT display. HSYNC is also used with $\overline{\text{FLN}}$ to reset the character line counter at the beginning of a character row (This will be covered in the discription of page 8 of the Video Logics).

Vertical Synchronization

VSYNC is generated by the VTAC to sychronize the top of the video frame. VSYNC goes to a one-shot in Chip 1K. the one-shot stretches the pulse and generates VDRIVE, $\overline{\text{VDRIVE}}$, and $\overline{\text{VSYNC}}$. VDRIVE goes to a counter on the CPU Board (page 8 CPU Logics) and is used to generate BLINKCLK. $\overline{\text{VDRIVE}}$ goes to the Monitor Board to designate the top of the video frame. $\overline{\text{VSYNC}}$ is used to force $\overline{\text{FLN}}$ low.

Line Attribute Data

The octal flip-flop at 1L (lower left of the page) is loaded directly by the Data Bus. When a character row is about to be displayed, Line Attribute Data is placed on the Data Bus. The Line Attribute Data gets strobed onto the flip-flops by $\overline{\text{SELIO48}}$. The flip-flops are three-state, but the enable line is tied to ground so the three-state feature is not used. The flip-flops hold the data on their outputs for the time it is needed while isolating the outputs from the Data Bus.

Scans Per Frame

The number of scans per frame is set by the firmware on Power-on/Reset according to SW4 on JMP7 (Page 13 of CPU Logics). If SW4 is installed, the microcode assumes the AC line frequency is 50Hz and sets 426 scans per frame. If SW4 is removed, the microcode assumes the AC line frequency is 60Hz and sets 354 scans per frame. Proper setting of this switch will help eliminate the appearance of "hum bars" from the screen. The extra scan lines in 50Hz Mode are blanked and used for delay time between vertical synch pulses.

The number of scans per frame can be altered by downloaded code by writing to I/O Port A0H.

Page 5 BUFFER ADDRESS GENERATOR

The Buffer Address Generator consists of a pair of 4-bit binary counters cascaded together and a pair of two-line to one-line multiplexers.

The outputs of the multiplexers address Buffer 1. At any time, the multiplexers either address Buffer 1 during the DMA load, or they address Buffer 1 for a data transfer to Buffer 2.

When $\overline{\text{XFER}}$ is high and $\overline{\text{BUSAKZ80}}$ is low, DMA has been acknowledged and the Buffer Control State Machine is ready to load characters into Buffer 1 (refer to Figure 7-2).

When the Buffer 1 loading process starts, $\overline{\text{SELIO60}}$ will go low and the counters 3L and 2L will be initialized by the Data Bus with Buffer 1's starting address. Both the lower and upper halves of Buffer 1 will be addressed the same. Only one half of the buffer will be write enabled at any given time. The signal R/$\overline{\text{W}}$BUF1L will go low and then high. This will be followed by R/$\overline{\text{W}}$BUF1U going low and then high. On the upward edge of R/$\overline{\text{W}}$BUF1U, the counters will increment to the address of the next memory location in both halves of Buffer 1. This will continue until Buffer 1 is filled.

$\overline{\text{XFER}}$ will go low and stay low during the process of transferring data from Buffer 1 to Buffer 2. When $\overline{\text{XFER}}$ goes low, the lines H0 - H6 are gated through the multiplexers 5F and 5D instead of the counters. H0 - H6 are also directly wired to the address lines of Buffer 2 (see page 7). The outputs of the multiplexers are used to address Buffer 1. While $\overline{\text{XFER}}$ is low, Buffer 1 can be read but not written to. R/$\overline{\text{W}}$BUF2 goes low during each write to Buffer 2. R/$\overline{\text{W}}$BUF2 goes high during a CHARLD and the lines H0 - H6 get incremented at this time.

EXPLANATION OF VIDEO LOGICS

Page 6 VIDEO CHARACTER BUFFER 1

This page shows the Buffer 1 address lines from the previous page going to the lower and upper halves of Buffer 1.

During the load sequence of Buffer 1, data will come into the octal flip-flop at 5L (upper left of the page).

First, a character data byte will be loaded into the flip-flops. The flip-flops will latch this byte. R/$\overline{\text{W}}$BUF1L will write enable the lower half of Buffer 1 (4J) while at the same time, the signal will enable the three-state gate buffer (5H) to pass the character data to 4J's data bus.

Next, R/$\overline{\text{W}}$BUF1L will go high and an attribute byte will be loaded onto the Data Bus and be latched by the octal flip-flop. Then R/$\overline{\text{W}}$BUF1U will go low to write enable the upper half of Buffer 1 (4L) while at the same time, the signal will enable the three-state gate buffer (5J) to pass the attribute data to 4L's data bus.

R/$\overline{\text{W}}$BUF1U will then go high and increment the Buffer 1 address count and the cycle will repeat for the next character in the row.

After the character row is finished being read into Buffer 1, it will be time to transfer the data to Buffer 2. During this time, $\overline{\text{XFER}}$ will be held low (as described in the explanation of page 5) and Buffer 1 and Buffer 2 will be addressed by the VTAC. The three-state gates will go to their high impedance states and the contents of Buffer 1 will be shifted to Buffer 2 via the Buffer 1 data bus (B1D0 - B1D7 and B1D10 - B1D17).

Page 7 VIDEO CHARACTER BUFFER 2

Logic Page 7 is similar to Page 6. The main difference is that all 16 bits of character and attribute data can be written to Buffer 2 at the same time.

R/$\overline{\text{W}}$BUF2 enables the data to pass through the gate buffers 5E and 5G when the signal goes low. At the same time, R/$\overline{\text{W}}$BUF2 write enables Buffer 2. Chip 4H is the character half of Buffer 2 and Chip 4F is the attribute half.

Buffer 2 continues to be filled while $\overline{\text{XFER}}$ is held low. R/$\overline{\text{W}}$BUF2 goes low during a write and then high while H0 - H6 (Buffer 2's address lines) are incremented by the VTAC. Since all 16 bits of data can be written to Buffer 2 at one time, Buffer 2 will be filled twice as fast as Buffer 1 was filled.

Buffer 2 is also the output buffer to the Character Generator. As Buffer 2 is being loaded, the character data is also being sent out to the Character Generator. When Buffer 2 is filled, the first scan line, if it contains any dots, is being shot to the screen by the electron beam.

CHARDOTCK is used to latch each character (with attribute) onto the Character Generator's address bus. After one scan line has been displayed, Buffer 2's address counters, H0 - H6 will be reset and incremented on each CHARLD as the next scan line of character data is displayed on the screen. The Buffer 2 address counters will cycle in this manner until all scan lines have been displayed for the character row.

Figure 7-2 shows that Buffer 2 is read enabled at all times when data is not being transferred to the buffer. Display is in fact occurring during almost the entire field of Figure 7-2. $\overline{\text{FLN}}$ goes low one CHARLD pulse before $\overline{\text{XFER}}$ goes low. After Buffer 2 loads (and the first scan line is "painted" on the screen) $\overline{\text{FLN}}$ goes high one CHARLD pulse before $\overline{\text{XFER}}$ goes high. The next scan lines will be displayed before, during, and after the load sequence of Buffer 1.

EXPLANATION OF VIDEO LOGICS

Page 8 CHARACTER GENERATOR

Just before each character line is displayed on the screen, FLN clocks the line attribute codes and the scan line count code into the octal D flip-flop at 2H (upper left of page).

The scan line count code goes to a comparator (4E). Here the code is compared with the current scan line count from 3B (middle left). The current scan count has the least significant bit dropped. The dropping of the LSB doubles the scan line count. The preloaded code for the scan line count is 2n + 2 where n is the 4-bit binary coded number fed into the comparator. Therefore, the scan line count will always be an even number.

When the A=B output of the comparator is true (high) an AND gate at 3H waits for the LSB of counter 3B to go high. When both of these conditions become true, LLN (the signal for the last scan line of the character row) is generated. On the next BLANK signal (from the VTAC circuit page 4), FLN and $\overline{\text{FLN}}$ are generated by the flip-flop at 1H (center of page). FLN strobes the new line attribute byte into the 8 X D flip-flop chip at 2H. FLN and HSYNC are ANDed together to reset line counter 3B for the beginning of the next character row.

Line attributes and some character attributes are handled by the three-state PROM at 4D. Subscripts and Superscripts require extra scan lines with offsets on the affected character rows.

The Attribute PROM also controls single and double underlining. Every character in the Character Generator has a bottom scan line of solid dots. This line is normally not in the range of the video display. However, the Attribute PROM can generate an offset and put this scan line within the display range of any character. In the case of double underlining, the Attribute PROM does this twice for affected characters.
The outputs of the Attribute PROM go to the four least significant address bits of the Character Generator PROM. These LSBs of the address are incremented each scan line since the characters are arranged to be sliced. The upper address bits are changed according to the outputs of the lower half of Character Buffer 2. The character row is thus displayed as follows:

1st line, 1st char.; 1st line, 2nd char.; . . . 1st line, last char.
2nd line, 1st char.; 2nd line, 2nd char.; . . . 2nd line, last char.
. . . until finally . . .
last line, last character;

Page 9 VIDEO OUTPUT

The outputs from the Character Generator go to a pair of 4-bit, bi-directional shift registers 2B and 2C (lower left of the page). The registers are cascaded to make an eight-bit register. The registers are hard-wired to shift the MSB out first and the LSB out last.

The eight bits (VD0 - VD7) are loaded into the registers when CHARLD is high. When CHARLD goes low, shifting is enabled. DOTCLK strobes the serial bits from the shift register pair. The output (Pin 12 of 2C) is the dot signal that drives the video output.

All the character attributes (except for single and double underlining) arrive at the 8 X D Flip-Flop at 3C (upper left of page) from the upper (attribute) half of Buffer 2.

Reverse Video

During normal display mode, the dots will pass through an XOR gate at 5C (center of page) without being affected. When reverse video characters are being displayed, the RVRS signal will be gated through Flip-Flop Chip 3C (upper left) and into the other leg of the XOR gate. This will cause the XOR gate to invert the output of the shift register.

Blink Attribute

The BLINK Attribute enables BLINKCLK through an AND gate at 1J. The signal proceeds through an XOR gate at 2A. The other leg of the XOR gate is tied to RVRS. BLINK toggles reverse video on and off in time with BLINKCLK. The result is that if BLINK is active, blinking normal characters will blink 180 degrees out of phase with blinking reverse video characters.

Blanking

$\overline{\text{BLANK}}$ comes into Flip-Flop Chip 3C and is generated by the VTAC on page 4. This signal blanks areas on the screen which are outside the display boundaries as well as the video output during flyback.

$\overline{\text{BLANKA}}$ from 5A is a composite of LLNBLNK, LLN, and RVRS and $\overline{\text{BLANK}}$ at 3C. $\overline{\text{BLANKA}}$ is always active when incoming $\overline{\text{BLANK}}$ is active. In addition, $\overline{\text{BLANKA}}$ will be active if LLBLNK is active during the last scan line when the VT4 is in reverse video mode. (The microcode always takes care of this when the VT4 is in VT3 emulation mode.)

Bold Characters

BOLD increases the video output. Everything for normal video output must be enabled in order for BOLD to have any effect. BOLD gets pipelined straight through the flip-flops at 3C (upper left) and 5A (upper right).

Shaded characters

SHAD gets pipelined straight through the flip-flops at 3C and 5A and becomes SHADE. SHADE is gated to the video output and generates a video signal over all the dark areas of displayed shaded characters. The SHADE output is variable by potentiometer R16. The range of R16 can turn shading on so high that it will wash out the affected characters, or R16 can turn shading on so low that it will not show on the screen at all.

EXPLANATION OF VIDEO LOGICS

STANDARD MICROSYSTEMS CORPORATION

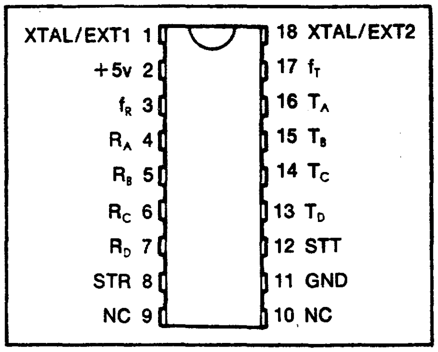
COM 8116
COM 8116T

Dual Baud Rate Generator
Programmable Divider

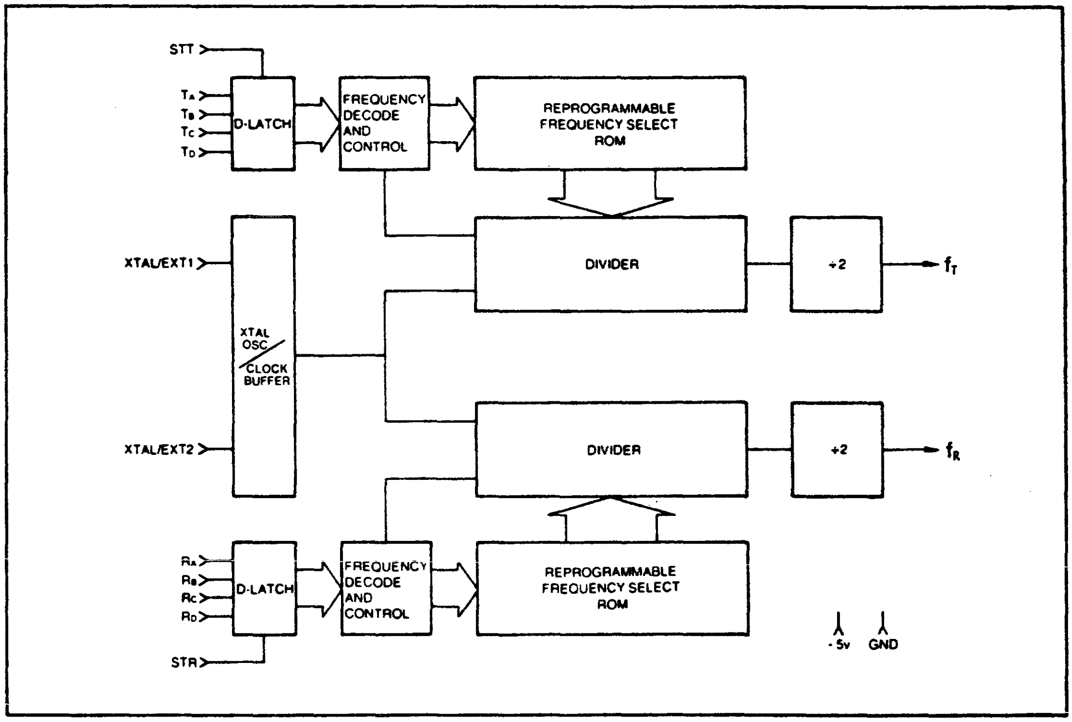
FEATURES

- □ On chip crystal oscillator or external frequency input
- □ Single +5v power supply
- □ Choice of 2 x 16 output frequencies
- □ 16 asynchronous/synchronous baud rates
- □ Direct UART/USRT/ASTRO/USYNRT compatibility
- □ Full duplex communication capability
- □ Re-programmable ROM via CLASP® technology allows generation of other frequencies
- □ TTL, MOS compatibility
- □ Compatible with COM 5016

PIN CONFIGURATION



BLOCK DIAGRAM



Reprinted with permission from STANDARD MICROSYSTEMS CORPORATION.

General Description

The Standard Microsystem's COM 8116 is an enhanced version of the COM 5016 Dual Baud Rate Generator. It is fabricated using SMC's patented COPLAMOS® and CLASP® technologies and employs depletion mode loads, allowing operation from a single +5v supply.

The standard COM 8116 is specifically dedicated to generating the full spectrum of 16 asynchronous/synchronous data communication frequencies for 16X UART/USRT devices. A large number of the frequencies available are also useful for 1X and 32X ASTRO/USYNRT devices.

The COM 8116 features an internal crystal oscillator which may be used to provide the master reference frequency. Alternatively, an external reference may be supplied by applying complementary TTL level signals to pins 1 and 18. Parts suitable for use only with an external TTL reference are marked COM 8116T. TTL outputs used to drive the COM 8116 or COM 8116T XTAL/EXT inputs should not be used to drive other TTL inputs, as noise immunity may be compromised due to excessive loading.

The output of the oscillator/buffer is applied to the dividers for generation of the output frequencies f_T, f_R. The dividers are capable of dividing by any integer from 6 to 2^{19} + 1, inclusive. If the divisor is even, the output will be square; otherwise the output will be high longer than it is low by one fx clock period.

Each of the two divisor ROMs contains 16 divisors, each 19 bits wide, and is fabricated using SMC's unique CLASP® technology allowing up to 32 different divisors on custom parts. This process permits reduction of turn-around time for ROM patterns. Each group of four divisor select bits is held in an externally strobed data latch. The strobe input is level sensitive: while the strobe is high, data is passed directly through to the ROM. Initiation of a new frequency is effected within 3.5µs of a change in any of the four divisor select bits (strobe activity is not required). The divisor select inputs have pull-up resistors; the strobe inputs do not.

Description of Pin Functions

Pin No.	Symbol	Name	Function
1	XTAL/EXT1	Crystal or External Input 1	This input is either one pin of the crystal package or one polarity of the external input.
2	V_{CC}	Power Supply	+5 volt supply
3	f_R	Receiver Output Frequency	This output runs at a frequency selected by the Receiver divisor select data bits.
4-7	R_A, R_B, R_C, R_D	Receiver-Divisor Select Data Bits	The logic level on these inputs, as shown in Table 1, selects the receiver output frequency, f_P.
8	STR	Strobe-Receiver	A high level input strobe loads the receiver data (R_A, R_B, R_C, R_D) into the receiver divisor select register. This input may be strobed or hard-wired to a high level.
9	NC	No Connection	
10	NC	No Connection	
11	GND	Ground	Ground
12	STT	Strobe-Transmitter	A high level input strobe loads the transmitter data (T_A, T_B, T_C, T_D) into the transmitter divisor select register. This input may be strobed or hard-wired to a high level.
13-16	T_D, T_C, T_B, T_A	Transmitter-Divisor Select Data Bits	The logic level on these inputs, as shown in Table 1, selects the transmitter output frequency, f_T.
17	f_T	Transmitter Output Frequency	This output runs at a frequency selected by the Transmitter divisor select data bits.
18	XTAL/EXT2	Crystal or External Input 2	This input is either the other pin of the crystal package or the other polarity of the external input.

COM8116, COM8116T, COM8126, COM8126T
COM8136, COM8136T, COM8146, COM8146T

Baud Rate Generator Output Frequency Options

Table 1. (16X clock)

CRYSTAL FREQUENCY = 5.0688 MHz

Tr'mit/Receive Address				Baud	Theoretical Frequency	Actual Frequency	Percent	Duty Cycle	
D	C	B	A	Rate	16X Clock	16X Clock	Error	%	Divisor
0	0	0	0	50	0.8 KHz	0.8 KHz	—	50/50	6336
0	0	0	1	75	1.2	1.2	—	50/50	4224
0	0	1	0	110	1.76	1.76	—	50/50	2880
0	0	1	1	134.5	2.152	2.1523	0.016	50/50	2355
0	1	0	0	150	2.4	2.4	—	50/50	2112
0	1	0	1	300	4.8	4.8	—	50/50	1056
0	1	1	0	600	9.6	9.6	—	50/50	528
0	1	1	1	1200	19.2	19.2	—	50/50	264
1	0	0	0	1800	28.8	28.8	—	50/50	176
1	0	0	1	2000	32.0	32.081	0.253	50/50	158
1	0	1	0	2400	38.4	38.4	—	50/50	132
1	0	1	1	3600	57.6	57.6	—	50/50	88
1	1	0	0	4800	76.8	76.8	—	50/50	66
1	1	0	1	7200	115.2	115.2	—	50/50	44
1	1	1	0	9600	153.6	153.6	—	48.52	33
1	1	1	1	19.200	307.2	316.8	3.125	50/50	16

Table 2. (16X clock)

CRYSTAL FREQUENCY = 4.9152 MHz

Tr'mit/Receive Address				Baud	Theoretical Frequency	Actual Frequency	Percent	Duty Cycle	
D	C	B	A	Rate	16X Clock	16X Clock	Error	%	Divisor
0	0	0	0	50	0.8 KHz	0.8 KHz	—	50/50	6144
0	0	0	1	75	1.2	1.2	—	50/50	4096
0	0	1	0	110	1.76	1.7589	-0.01	*	2793
0	0	1	1	134.5	2.152	2.152	—	50/50	2284
0	1	0	0	150	2.4	2.4	—	50/50	2048
0	1	0	1	300	4.8	4.8	—	50/50	1024
0	1	1	0	600	9.6	9.6	—	50/50	512
0	1	1	1	1200	19.2	19.2	—	50/50	256
1	0	0	0	1800	28.8	28.7438	-0.19	*	171
1	0	0	1	2000	32.0	31.9168	-0.26	50/50	154
1	0	1	0	2400	38.4	38.4	—	50/50	128
1	0	1	1	3600	57.6	57.8258	0.39	*	85
1	1	0	0	4800	76.8	76.8	—	50/50	64
1	1	0	1	7200	115.2	114.306	-0.77	*	43
1	1	1	0	9600	153.6	153.6	—	50/50	32
1	1	1	1	19,200	307.2	307.2	—	50/50	16

Table 3. (32X clock)

CRYSTAL FREQUENCY = 5.0688 MHz

Tr'mit/Receive Address				Baud	Theoretical Frequency	Actual Frequency	Percent	Duty Cycle	
D	C	B	A	Rate	32X Clock	32X Clock	Error	%	Divisor
0	0	0	0	50	1.6 KHz	1.6 KHz	—	50/50	3168
0	0	0	1	75	2.4	2.4	—	50/50	2112
0	0	1	0	110	3.52	3.52	—	50/50	1440
0	0	1	1	134.5	4.304	4.306	.06	*	1177
0	1	0	0	150	4.8	4.8	—	50/50	1056
0	1	0	1	200	6.4	6.4	—	50/50	792
0	1	1	0	300	9.6	9.6	—	50/50	528
0	1	1	1	600	19.2	19.2	—	50/50	264
1	0	0	0	1200	38.4	38.4	—	50/50	132
1	0	0	1	1800	57.6	57.6	—	50/50	88
1	0	1	0	2400	76.8	76.8	—	50/50	66
1	0	1	1	3600	115.2	115.2	—	50/50	44
1	1	0	0	4800	153.6	153.6	—	*	33
1	1	0	1	7200	230.4	230.4	—	50/50	22
1	1	1	0	9600	307.2	316.8	3.125	50/50	16
1	1	1	1	19,200	614.4	633.6	3.125	50/50	8

As used with the VT4

OUTPUT FREQUENCY OPTIONS

Dash Number		
Table 1	Table 2	Table 3
STD	-5	-6
STD	-5	-6

*When Duty Cycle s not exactly 50%, it is 50% ± 10%.

STANDARD MICROSYSTEMS
CORPORATION

CRT 5027
CRT 5037
CRT 5057

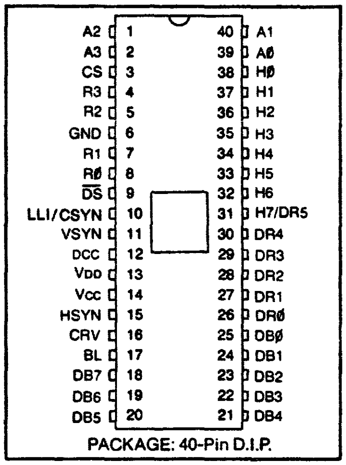
μPC FAMILY

CRT Video Timer and Controller VTAC®

FEATURES

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

PIN CONFIGURATION



PACKAGE: 40-Pin D.I.P.

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

GENERAL DESCRIPTION

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

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

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

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

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

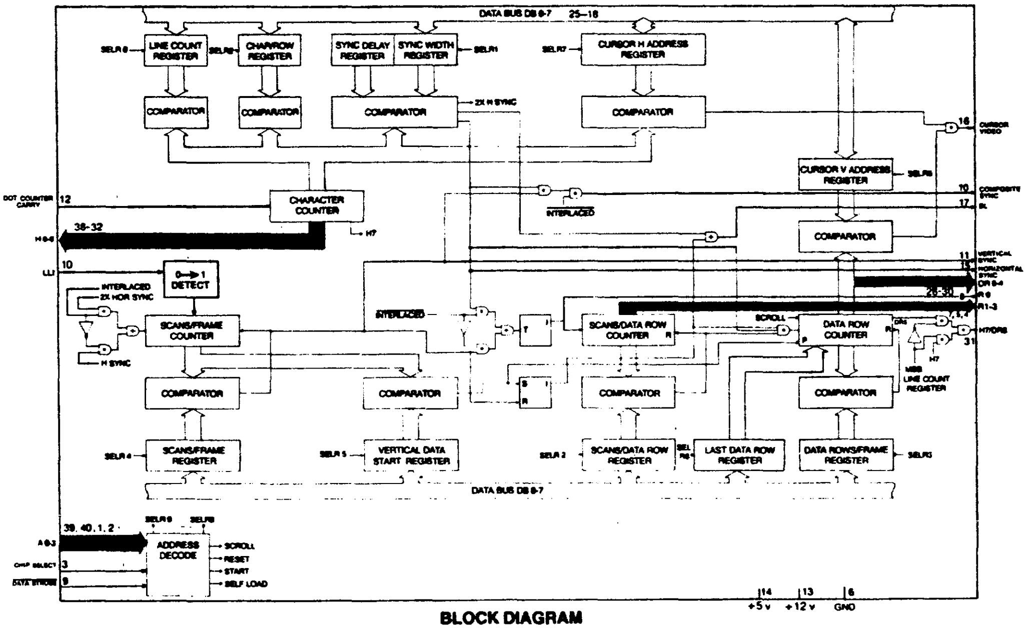
The CRT 5057 provides the ability to lock a CRT's vertical refresh rate, as controlled by the VTAC's® vertical sync pulse, to the 50 Hz or 60 Hz line frequency thereby eliminating the so called "swim" phenomenon. This is particularly well suited for European system requirements. The line frequency waveform,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

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

Reprinted with permission from STANDARD MICROSYSTEMS CORPORATION.

Description of Pin Functions

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



BLOCK DIAGRAM

Operation

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

Horizontal Formatting:

Characters/Data Row — A 3 bit code providing 8 mask programmable character lengths from 20 to 132. The standard device will be masked for the following character lengths: 20, 32, 40, 64, 72, 80, 96, and 132.

Horizontal Sync Delay — 3 bits assigned providing up to 8 character times for generation of "front porch".

Horizontal Sync Width — 4 bits assigned providing up to 15 character times for generation of horizontal sync width.

Horizontal Line Count — 8 bits assigned providing up to 256 character times for total horizontal formatting.

Skew Bits — A 2 bit code providing from a 0 to 2 character skew (delay) between the horizontal address counter and the blank and sync (horizontal, vertical, composite) signals to allow for retiming of video data prior to generation of composite video signal. The Cursor Video signal is also skewed as a function of this code.

Vertical Formatting:

Interlaced/Non-interlaced — This bit provides for data presentation with odd/even field formatting for interlaced systems. It modifies the vertical timing counters as described below. A logic 1 establishes the interlace mode.

Scans/Frame — 8 bits assigned, defined according to the following equations: Let X = value of 8 assigned bits.

1) in interlaced mode—scans/frame = 2X + 513. Therefore for 525 scans, program X = 6 (00000110). Vertical sync will occur precisely every 262.5 scans, thereby producing two interlaced fields.
Range = 513 to 1023 scans/frame, odd counts only.

2) in non-interlaced mode—scans/frame = 2X + 256. Therefore for 262 scans, program X = 3 (00000011).
Range = 256 to 766 scans/frame, even counts only.

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

Vertical Data Start — 8 bits defining the number of raster scans from the leading edge of vertical sync until the start of display data. At this raster scan the data row counter is set to the data row address at the top of the page.

Data Rows/Frame — 6 bits assigned providing up to 64 data rows per frame.

Last Data Row — 6 bits to allow up or down scrolling via a preload defining the count of the last displayed data row.

Scans/Data Row — 4 bits assigned providing up to 16 scan lines per data row.

Additional Features

Device Initialization:

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

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

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

Control Registers Programming Chart

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

Characters/Data Row:

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

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

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

Skew Bits

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

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

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

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

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

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

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

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

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

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

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

Scans/Data Row: Interlace Mode

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

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

Non-Interlace Mode

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

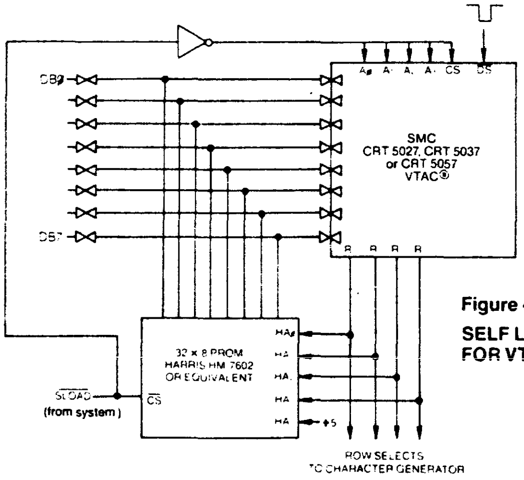


Figure 4.

SELF LOADING SCHEME FOR VTAC® SET-UP

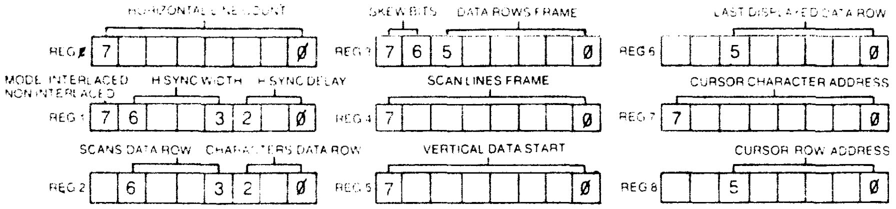
Register Selects/Command Codes

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

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

TABLE 1

BIT ASSIGNMENT CHART


HORIZONTAL LINE COUNT
REG Ø
7
Ø
SKEW BITS
DATA ROWS FRAME
REG 3
7
6
5
Ø
LAST DISPLAYED DATA ROW
REG 6
5
Ø
MODE INTERLACED
NON INTERLACED
H SYNC WIDTH
H SYNC DELAY
REG 1
7
6
3
2
Ø
SCAN LINES FRAME
REG 4
7
Ø
CURSOR CHARACTER ADDRESS
REG 7
7
Ø
SCANS DATA ROW
CHARACTERS DATA ROW
REG 2
6
3
2
Ø
VERTICAL DATA START
REG 5
7
Ø
CURSOR ROW ADDRESS
REG 8
5
Ø