VL-VID-64
VL-VID-80

Video Display Generator

STD VID-64/80
Video Display Generator

Table of Contents

VersaLogic
3888 Stewart Road
Eugene, Oregon 97402
(503) 485-8575

PN 2850 Rev. 2/84

MVID80

Introduction

The VID 64/80 is a versatile memory-mapped video display board for the STD BUS.

Features of the VID 64/80 such as flexible screen format, programmable character size, alphanumeric and graphic characters, and 2K on-board RAM make it easily adapted to a wide variety of display requirements.

Regardless of the application, the features, flexibility and dependability of the VID 64/80 make it a major asset in any STD BUS system.

Overview

Although specific information about each of the VID 64/80 functions can be found elsewhere in this manual, this section presents general information about the board and its capabilities.

The VID 64/80 is a "memory-mapped" video board, meaning that the RAM on the VID 64/80 card (which holds the characters currently being displayed) is part of the memory map of the computer system. The VID 64/80's 2K on-board RAM can be read or written like any other system RAM. Each RAM location corresponds to one character position on the screen. Memory-mapped video boards allow much faster display updating and more flexibility in handling special display formats than I/O mapped or terminal type displays. The switch selectable addressing on the VID 64/80 board allows it to be placed on any 2K boundary in the 64K memory map.

The heart of the VID 64/80 is a video controller chip (CRTC). The CRTC generates the "sync" signals that control scanning of the screen as well as reading the on-board RAM to continuously refresh the display image. The CRTC can be programmed for the desired number of characters per line and lines per screen.

In addition to programming the CRTC for the number of characters per line (to the 64 or 80 character maximum) and the number of lines per screen, the VID 64/80 provides manual controls (at the board edge) for horizontal and vertical centering of the display on the monitor. There is also a programmable control that allows an increased space between lines for an easier to read display. All of these features allow the VID 64/80 to be completely tailored to the monitor in use as well as the requirements of the application.

The VID 64/80 also allows a great deal of flexibility in its character set. In addition to the normal ASCII set of alphanumeric characters, making the upper bit high will display any character in reverse (black on white instead of white on black) for emphasis. The 32 graphic characters contain symbols such as thick and thin vertical and horizontal lines, right angles, diagonals and semi-solids that can be combined to form shapes, line drawings, boxes, bar graphs, etc. Graphic characters are also available in reverse for additional display combinations. The VID 64/80's greatest flexibility in character generation is the ability to remove the VersaLogic character generator ROM and substitute a 2716 PROM programmed with a custom character set. After including the standard alphanumeric characters, there is room for 160 special purpose characters- enough for any application.

Another feature of the VID 64/80 is the ability to expand the entire display to 2X or 4X the normal size. This results in a large, easily readable display that is ideal for status displays, testing/adjusting the hardware at a distance, commercial message displays, etc. When this programmable feature is activated, any of the VID 64/80's characters written to the display will be displayed in the expanded size.

Connection to the STD BUS

The VID 64/80 board requires only +5V (regulated) power and is compatible with both 65/6800 and 8085/Z-80 type STD BUS systems. The VID 64/80 "W1" or "W2" jumper must be connected when it is used in Z-80 or 8085 systems (see Jumper Options).

When inserting the card in a STD BUS card cage, be certain that the card ejector (pin 1 edge of the card) is aligned in the same direction as other cards in the system. The VID 64/80 card edge has a key slot cut between pins 25 and 27. It is recommended that a matching key be installed in the motherboard connector to prevent the card from being installed upside down.

The VID 64/80 board should be inserted or removed from the STD BUS card cage only when the +5V power to the bus is off.

Output Connection

The output of the VID 64/80 board is a standard composite video signal, designed to drive a 75 Ohm load. Connection between the VID 64/80 and the video monitor(s) should be made with common 50-75 Ohm coaxial cable (such as RG59, RG58 or RG174).

After soldering the two pin socket connector (included) to the coaxial cable it is easily connected to the two-pin header on the VID 64/80 board. However, when the cable is connected to the VID 64/80 board it must be oriented correctly. The signal lead of the coaxial cable must connect to pin 1 (nearest the card ejector) while the ground braid connects to pin 2.

Normally the cable from the VID 64/80 is not connected directly to the video monitor, but is run a short distance to a panel mounted connector. This provides convenient external access as well as a sturdy connection point for the video monitor cable.

Note- The VID 64/80 also includes non-composite TTL level signal outputs to drive OEM CRT display products. Refer to the VID 64/80 schematic or contact the factory for further information on these signals.

Board Address

The VID 64/80 board occupies a 2K byte block of memory which can start on any 2K memory boundary. This 2K block contains both the on-board video refresh RAM, and the video controller chip registers.

The location of the video board is usually determined by the requirements of the system monitor or video driver software which will be used with the board. Consult the software or system documentation involved for further information.

After determining the desired starting address for the VID 64/80 board, the binary values of the upper five address bits (A15-A11) are set into the switches near the bottom of the board. Each switch is set to the left (off) for a "0" and to the right (on) for a "1" value. The following chart notes switch settings for all possible locations in memory.

The three remaining switches should normally be set as follows: MX and AM - off, RV - on. See Switch Options for further information.

VID 64/80 Board Address Switch Settings

Starting Address (Hex.)	A15	A14	A13	A12	A11	AUX CNTL Port Address
0000	0	0	0	0	0	07F8
0800	0	0	0	0	1	0FF8
1000	0	0	0	1	0	17F8
1800	0	0	0	1	1	1FF8
2000	0	0	1	0	0	27F8
2800	0	0	1	0	1	2FF8
3000	0	0	1	1	0	37F8
3800	0	0	1	1	1	3FF8
4000	0	1	0	0	0	47F8
4800	0	1	0	0	1	4FF8
5000	0	1	0	1	0	57F8
5800	0	1	0	1	1	5FF8
6000	0	1	1	0	0	67F8
6800	0	1	1	0	1	6FF8
7000	0	1	1	1	0	77F8
7800	0	1	1	1	1	7FF8
8000	1	0	0	0	0	87F8
8800	1	0	0	0	1	8FF8
9000	1	0	0	1	0	97F8
9800	1	0	0	1	1	9FF8
A000	1	0	1	0	0	A7F8
A800	1	0	1	0	1	AFF8
B000	1	0	1	1	0	B7F8
B800	1	0	1	1	1	BFF8
C000	1	1	0	0	0	C7F8
C800	1	1	0	0	1	CFF8
D000	1	1	0	1	0	D7F8
D800	1	1	0	1	1	DFF8
E000	1	1	1	0	0	E7F8
E800	1	1	1	0	1	EFF8
F000	1	1	1	1	0	F7F8
F800	1	1	1	1	1	FFF8

Note: 0 = off, 1 = on

Software Requirements

In order to use the VID 64/80, some type of interface software or "driver" routine is necessary. Normally such a routine accepts single characters from the system and displays them on the screen one character after the next. This "video driver" routine is responsible for handling cursor movement and other chores such as scrolling the display when it becomes full.

A relatively simple video driver routine for 8085/Z80 systems is included as an appendix to this manual. It can be used as is for many applications, or the assembly language routines can be modified or expanded for more complex applications. As written, this routine initializes the board to a desired format (see page 2 of the listing), and implements the following control functions:

Function	Control code (hex)
Carriage return	0D
Line feed (cursor down)	0A
Clear screen	0C
Cursor home	01
Backspace (cursor left)	08
Cursor up	1A
Cursor right	06

Systems which use the VersaLogic Z80 Smart Card as the system processor already have an extended video driver routine included in the on-board firmware and do not need any additional software support for the STD VID 64/80 board. This video driver routine which is built into the Z80 Smart Card includes the above control functions, plus normal or reverse character toggle, alpha or graphic character set toggle, and character size change commands.

If one of these existing video driver routines will not be used, the following can be helpful in writing custom driver routines.

INITIALIZATION: Before the VID 64/80 can display anything, a number of parameters must be written to the controller chip. These parameters describe how many characters are in a line, how many lines per screen, etc.

Rather than discuss the calculations involved in setting these parameters, refer to the following Parameter Tables which allow the VID 64/80 to be initialized for a number of typical formats. For complete details on parameter calculation, refer to the 6845 Data Sheet elsewhere in this manual. As noted in these tables, the 28 line per screen format should normally be used for displays involving graphic characters. Formats with less than 28 lines have extra blank scan lines inserted between character rows which don't allow graphic characters to interconnect for contiguous lines, shapes, etc.

After selecting a screen format, a short routine is used to write the parameters to the VID 64/80 board. A typical routine contains a table of parameters and writes them to each register by 1), selecting the next register (using the CRTC Register Select port) and 2), writing the data (using the CRTC Data port) to the register.

Once the VID 64/80 board is initialized it will retain the programmed format until the power is removed or the board is re-initialized with another format. Initialization is not affected by system reset.

USE. In use as an output device, the driver routine must map each incoming character to its proper location on the display. Since each location on th display corresponds to a particular RAM location, this really amounts to writing each character to the proper RAM location (on the RAM board).

The starting address of the board (hex 9000 for example) always corresponds to the character in the upper left corner of the screen. Writing hex 41 (the ASCII code for an "A") to location 9000 would make an "A" appear in the upper left position of the screen. Location 9001 holds the next character on the line, etc., thru the 80th character (assuming an 80 character board initalized to an 80 character format). The 81st address (from the starting location) appears at the left edge of the screen on the second line. The video RAM is mapped in this pattern for the entire screen- left to right on each line from the first line at the top of the screen to the last line at the bottom of the screen. Note that the number of RAM locations per line corresponds to the number of characters per line that the board has been initialized for. Changing to larger characters (such as 40 characters per line instead of 80 characters/line) will alter the beginning address for each display line except the first.

Changing to a new format (i.e., character size) can be done at any time by re-initializing the control registers of the CRTC chip. Refer to the Parameter Table for common format parameters.

As noted in the following Character Code Chart there are 256 characters available for display. The first 128 characters are shown on the chart, while the second 128 are identical in a reversed (black on white rather than white on black) dot pattern. A standard 2716 PROM chip is used as for character generation allowing a custom character set to be substituted if required.

It should also be noted that two approaches can be taken to cursor generation. First, the cursor character (hex 7F) can be written into the desired screen location. Moving the cursor involves removing it from the current location (writing a space or other character in its place) and writing it at its new location. Alternately, the CRTC chip can be programmed to generate either a solid or blinking cursor using CRTC registers 10, 11,

14, and 16. After writing to registers 10 and 11 (usually 60 and 07 respectively) to enable the CRTC cursor, the address where the cursor will appear is written to registers 14 and 16. The cursor location address is mapped in the same way as the RAM display cells, but is relative to zero, rather than the starting address of the board.

Parameter Table

Common Screen Formats for 64 Character Version Boards

Register Number	64 X 28* Format	64 X 24 Format	64 X 22 Format	32 X 14** Format	16 X 7*** Format
0	5F	5F	5F	2F	17
1	40	40	40	20	10
2	46	46	46	24	13
3	01	01	01	01	01
4	1F (26)	1B (21)	19 (1E)	0F (12)	07 (08)
5	04 (00)	08 (06)	00 (02)	04 (08)	04 (18)
6	1C	18	16	0E	07
7	1C	18	16	0E	07
8	00	00	00	00	00
9	07	08	09	0F	1F
A	20	20	20	20	20
B	07	08	08	0F	1F
C	00	00	00	00	00
D	00	00	00	00	00
AUX CNTL	04	0C	0C	05	06

Parameter Table

Common Screen Formats for 80 Character Version Boards

Register Number	80 X 25* Format	80 X 24 Format	80 X 22 Format	40 X 14** Format	20 X 7*** Format
0	6F	6F	6F	37	1B
1	50	50	50	28	14
2	56	56	56	2C	17
3	01	01	01	01	01
4	1F (26)	1B (21)	19 (1E)	0F (12)	07 (08)
5	04 (00)	08 (06)	00 (02)	04 (08)	04 (18)
6	19	18	16	0E	07
7	1A	18	16	0E	07
8	00	00	00	00	00
9	07	08	09	0F	1F
A	20	20	20	20	20
B	07	08	08	0F	1F
C	00	00	00	00	00
D	00	00	00	00	00
AUX CNTL	04	0C	0C	05	06

Notes:

All numbers are hexadecimal

(XX) Use these parameters for 50 Hz (European) applications.

* Use for graphic character displays

** Double size characters

*** Quadruple size characters

Character Code Chart

VersaLogic Character Generator ROM (VID1)

The table below shows which character will be displayed for each ASCII (hex) value. Decimal equivalents are also listed. Each character listed can also be displayed in reverse (black on white) by setting bit 7 high (add 80 hex, 128 decimal).

Decimal	Hex	Char.	Decimal	Hex	Char.	Decimal	Hex	Char.
0	0		20	14		40	28	(
1	1		21	15		41	29	)
2	2		22	16		42	2A	*
3	3		23	17		43	2B	+
4	4		24	18		44	2C	,
5	5		25	19		45	2D	-
6	6		26	1A		46	2E	.
7	7		27	1B		47	2F	/
8	8		28	1C		48	30	0
9	9		29	1D		49	31	1
10	A		30	1E		50	32	2
11	B		31	1F		51	33	3
12	C		32	20		52	34	4
13	D		33	21	!	53	35	5
14	E		34	22	"	54	36	6
15	F		35	23	#	55	37	7
16	10		36	24	$	56	38	8
17	11		37	25	%	57	39	9
18	12		38	26	&	58	3A	:
19	13		39	27	'	59	3B	;

Decimal	Hex	Char.	Decimal	Hex	Char.	Decimal	Hex	Char.
60	3C	<	83	53	S	106	6A	j
61	3D	=	84	54	T	107	6B	k
62	3E	>	85	55	U	108	6C	l
63	3F	?	86	56	V	109	6D	m
64	40	@	87	57	W	110	6E	n
65	41	A	88	58	X	111	6F	o
66	42	B	89	59	Y	112	70	p
67	43	C	90	5A	Z	113	71	q
68	44	D	91	5B	[	114	72	r
69	45	E	92	5C	\	115	73	s
70	46	F	93	5D	]	116	74	t
71	47	G	94	5E	↑	117	75	u
72	48	H	95	5F	_	118	76	v
73	49	I	96	60	`	119	77	w
74	4A	J	97	61	a	120	78	x
75	4B	K	98	62	b	121	79	y
76	4C	L	99	63	c	122	7A	z
77	4D	M	100	64	d	123	7B	↓
78	4E	N	101	65	e	124	7C	→
79	4F	O	102	66	f	125	7D	←
80	50	P	103	67	g	126	7E	~
81	51	Q	104	68	h	127	7F	■
82	52	R	105	69	i			

Control Port Usage

This section details the use of the VID 64/80's control ports. Although most of this information is not required for normal video board use, it is included for those who want to use the VID 64/80 in unusual/custom applications.

AUX CNTL PORT

Referring to the Control Port Location section, it can be seen that the AUX CNTL port uses six bits (D0-D5) for various control functions. This is a "write only" port. Since the contents of this port cannot be read by the processor, it is usually best to keep a copy of this byte in RAM if many changes are expected to the port during use. Changes to the AUX CNTL port become effective immediately and remain until another update (write) to the port occurs. This port is cleared to zeros during power-up. Each AUX CNTL port function is discussed individually below.

D0 and D1 - CHARACTER SIZE. The lower two bits of the port allow characters to be displayed at normal (X1), double (X2), or quadruple (X4) size. For usable results, other parameters in the system (such as number of characters per line) must be changed when character size is altered. See the Parameter Tables of screen formats for further information.

D2 - SCREEN BLANKING. Setting this bit low blanks the contents of the display. Note that only the display output is affected; the characters in RAM are not altered and will re-appear when D2 is set to a one. Characters in VID 64/80 RAM may be altered while the screen is blanked. Note that the AUX CNTL port is cleared to zeros during power-up leaving the display in a blanked state. A "1" is usually written to the blanking control as part of the initialization process.

D3 - LINE SPACING. This bit is used to increase the blank space between character rows. Normally (when D3 = 0) a single scan line is present between character rows. This blank line is actually part of the dot pattern in the character generator ROM. Although the alphanumeric characters leave this scan line blank, the graphic characters do not respect this "dead space" and can therefore produce continuous vertical lines, shapes, etc. Graphic characters essentially operate with no spacing between lines. D3 should remain set to zero whenever graphic drawings will be used.

When only alphanumeric information is displayed, it is often desirable to have additional blank space between character rows for readability. The line spacing control bit (D3) can be used

in conjunction with the CRTC scan lines per row register (R9) to add these extra blank lines. When bit D3 is set to "1", any scan line greater than 7 will be blanked. Setting the CRTC register #9 to "8" would add one blank scan line. Setting register #9 to "9" would add two blank scan lines between character rows, etc. The line spacing bit (D3) may only be used with normal size (X1) characters.

D4 - VIDEO RAM LOCKOUT. This bit can be used to lock out all the R/W access to the on-board RAM by the system CPU. It can be used to prevent interference or "snow" which occurs in a rare number of cases during regular (non-display oriented) processing. The interference results from CPU address lines which change very slowly or do not change together. This can cause a momentary period where a valid video RAM address appears on the address lines and refresh of the video display is interrupted. If the RAM lockout bit (D4) is used, the video board will accept data into RAM only when D4 is set to "0".

D5 - VERTICAL SYNC ENABLE. This bit is used to enable/disable system interrupts at the start of every vertical sync (vertical retrace) period. Note that the "IRQ" pads on the VID 64/80 board must be jumpered before this interrupt can occur (see Jumper Options section). Vertical sync occurs approximately 60 times a second (50 times a second in Europe).

D6-D7 - These bits are not used by the AUX CNTL port and are ignored during writing.

AUX STATUS PORT

The AUX STATUS port contains two bits which can be read at any time. Reading this port does not interfere with the display refreshing process. Data bits D0-D5 are not used and may be any value when the port is read.

D6 - BLANKING. This bit will go low whenever horizontal or vertical blanking occurs. It is high whenver display refresh is taking place.

D7 - VERTICAL SYNC. This bit will be high for approximately 1 ms at the start of each vertical sync period. See AUX CNTL PORT - D5 and the Jumper Options section for interrupt capability with this signal. This status signal is available at all times and is not affected by the vertical sync interrupt enable control.

CRTC REGISTER SELECT PORT

This write-only port determines which of the CRTC registers will be accessed by the CRTC data port. The CRTC data port will continue to write into the same register until a new one is selected using the CRTC register select port.

CRTC DATA PORT

This port transfers data to the currently selected CRTC register. The CRTC registers are normally used only for initializing the VID 64/80 board. See Software Requirements- Initialization and the 6845 Data Sheet for further information.

Control Port Locations

The port locations are shown below relative to the AUX CNTL port address. The AUX CNTL port address can be found by referring to the table in the Board Address section or by adding 07F8 to the starting address of the board. Examples are shown below for a VID 64/80 board with a starting address of 9000 (hex).

Name	Address	Example	R/W
AUX CNTL	AUX CNTL +0	97F8	Write

D0 & D1 - Character size (see below)
D2 = 0 - To blank screen
D3 = 1 - For extra spacing between lines
D4 = 1 - To lockout video RAM access
D5 = 1 - To enable vertical sync interrupts
D6-D7 - Not used

Note: All bits are cleared to 0 during power-on.

Char. Size	D1	D0
X1	0	0
X2	0	1
X4	1	0
N/A	1	1

Name	Address	Example	R/W
AUX STATUS	AUX CNTL +0	97F8	Read

D6 = 0 - During horizontal and vertical blanking periods
D7 = 1 - For 1 ms at start of vertical sync.

Name	Address	Example	R/W
CRTC Register Select	AUX CNTL +4	97FC	Write

D0-D4 - Selects register for CRTC data port access
D5-D7 - Not used

Name	Address	Example	R/W
CRTC Register Data	AUX CNTL +6	97FE	Write

D0-D7 - Writes data to selected CRTC register

Jumper Options

VERT. RETRACE INTERRUPT. The pads marked "IRQ" (in the lower right corner of the board) can be jumpered to provide an interrupt request during the vertical sync period at the end of each screen refresh. Special applications may use this signal (which lasts 1 ms at the start of the retrace period) to update the display without the slight "snow" that normally occurs when writing to display RAM, or as a convenient 60 Hz (50 Hz Europe) interrupt for keyboard scanning, time-keeping, etc.

Once jumpered, the Vert. Retrace Interrupt can be enabled/disabled thru bit 5 in the AUX CNTL port. The Vert. Retrace Signal (as well as a vertical/horizontal blank signal) is also available in the AUX STATUS port.

WAIT CYCLE. Because the video controller chip on the VID 64/80 board requires a write cycle time greater than 400 ns., the memory write cycle must be lengthened when the VID 64/80 is used with 8085 or Z-80 STD BUS processor cards. To insert a wait cycle in each video controller chip access, jumper the pads marked "W1" for a Z-80 or "W2" for an 8085 in the lower right corner of the VID 64/80 board.

Switch Options

ADDRESS. Switches A15-A11 determine the starting address for the VID 64/80 board. Each address bit is set to a "0" (off) or "1" (on) as needed. See Board Address for further information.

MEMORY EXPANSION. The switch marked "EX" allows use of the VID 64/80 board in the secondary or expanded portion of the memory map in systems which use the MEMEX bus line. Most systems do not use this feature and require the MX switch set to the left (off) position. This setting enables the board when MEMEX is low. Note: A jumper option is available which allows the VID 64/80 to appear in both portions of expanded memory maps. Contact the factory for further details.

REVERSE VIDEO. The switch marked "RV" allows the entire screen image to be reversed (black on white) for special applications. For normal use the switch should be set to the right (on) position.

AIM-MATE SYSTEM USE. The switch marked "AM" allows the VID 64/80 to be used with the AIM-Mate expansion series for the AIM 65 computer. When used in AIM-Mate systems the switch should be set to the right (on). For use in all other STD BUS systems, this switch should be set to the left (off).

MOTOROLA Semiconductors
3501 ED BLUESTEIN BLVD., AUSTIN, TEXAS 78721

MC6845

Advance Information

CRT CONTROLLER (CRTC)

The MC6845 CRT Controller performs the interface to raster scan CRT displays. It is intended for use in processor-based controllers for CRT terminals in stand-alone or cluster configurations.

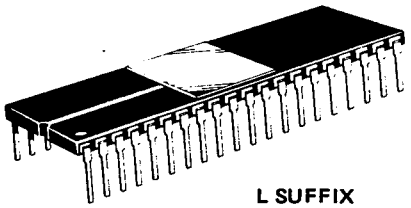
The CRTC is optimized for hardware/software balance in order to achieve integration of all key functions and maintain flexibility. For instance, all keyboard functions, R/W, cursor movements, and editing are under processor control; whereas the CRTC provides video timing and Refresh Memory Addressing.

- Applications include "glass-teletype," smart, programmable, intelligent CRT terminals; video games; information display.
- Alphanumeric, semi-graphic, and full graphic capability.
- Fully programmable via processor data bus. Can generate timing for almost any alphanumeric screen density, e.g. 80 x 24, 72 x 64, 132 x 20, etc.
- Single +5 volt supply. TTL/6800 compatible I/O.
- Hardware scroll (paging or by line or by character)
- Compatible with CPU's and MPU's which provide a means for synchronizing external devices.
- Cursor register and compare circuitry.
- Cursor format and blink are programmable.
- Light pen register.
- Line buffer-less operation. No external DMA required. Refresh Memory is multiplexed between CRTC and MPU.
- Programmable interlace or non-interlace scan.
- 14-bit wide refresh address.

MOS

(N-Channel, Silicon-Gate)

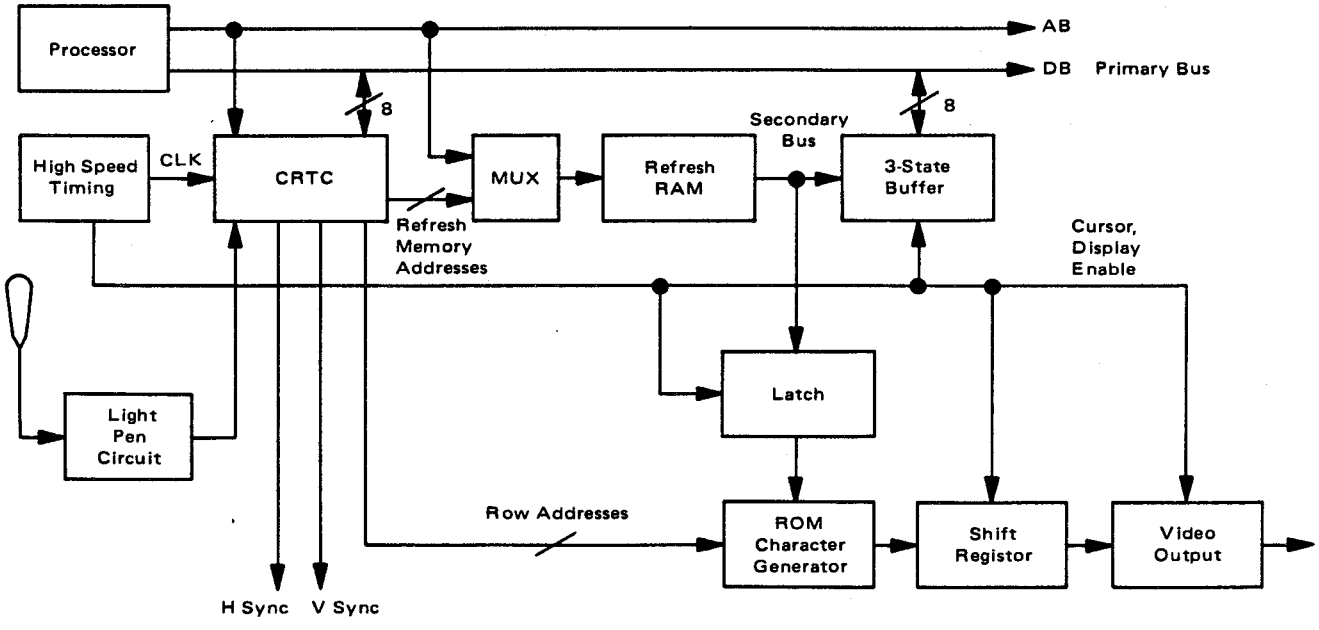
CRT CONTROLLER (CRTC)



L SUFFIX
CERAMIC PACKAGE
CASE 715

NOT SHOWN:
P SUFFIX
PLASTIC PACKAGE
CASE 711

FIGURE 1 – TYPICAL CRT CONTROLLER APPLICATION



AD1-465
(Replaces NP-67)

SYSTEM BLOCK DIAGRAM DESCRIPTION

As shown in Figure 1, the primary function of the CRTC is to generate refresh addresses (MA0-MA13), row selects (RA0-RA4), and video monitor timing (HSYNC, VSYNC) and Display Enable. Other functions include an internal cursor register which generates a Cursor output when its contents compare to the current Refresh Address. A light-pen strobe input signal allows capture of Refresh Address in an internal light pen register.

All timing in the CRTC is derived from the Clk input. In alphanumeric terminals, this signal is the character rate. Character rate is divided down from video rate by external High Speed Timing when the video frequency is greater than 3 MHz. Shift Register, Latch, and MUX Control signals are also provided by external High Speed Timing.

The processor communicates with the CRTC through a buffered 8-bit Data Bus by reading/writing into the 18-register file of the CRTC.

The Refresh Memory address is multiplexed between the Processor and CRTC. Data appears on a Secondary Bus which is buffered from the processor Primary Bus. A number of approaches are possible for solving contentions for the Refresh Memory.

1. Processor always gets priority.
2. Processor gets priority access anytime, but can be synchronized by an interrupt to perform accesses only during horizontal and vertical retrace times.
3. Synchronize processor by memory wait cycles.
4. Synchronize processor to character rate (See Figure 2). The 6800 MPU family lends itself to this configuration because it has constant cycle lengths. This method provides zero burden on the processor because there is never a contention for memory. All accesses are *"transparent."*

The secondary data bus concept in no way precludes using the Refresh RAM for other purposes. It looks like any other RAM to the Processor. For example, using Approach 4, a 64K byte RAM Refresh Memory could perform refresh and program storage functions transparently.

MAXIMUM RATINGS

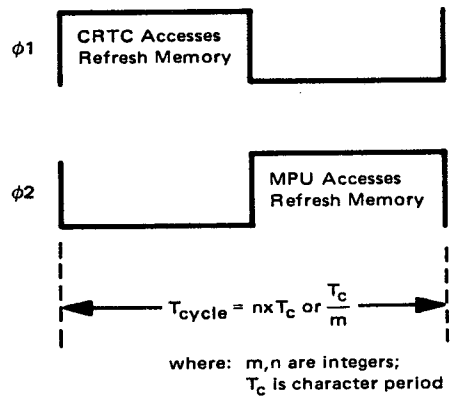
Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}*	-0.3 to +7.0	Vdc
Input Voltage	V_{in}*	-0.3 to +7.0	Vdc
Operating Temperature Range	T_A	0 to +70	°C
Storage Temperature Range	T_{stg}	-55 to +150	°C

*With respect to V_{SS} (Gnd).

RECOMMENDED OPERATING CONDITIONS

Characteristics	Symbol	Min	Typ	Max	Unit
Supply Voltage	V_{CC}	4.75	5.0	5.25	Vdc
Input Low Voltage	V_{IL}	-0.3	–	0.8	Vdc
Input High Voltage	V_{IH}	2.0	–	V_{CC}	Vdc

FIGURE 2 – TRANSPARENT REFRESH MEMORY CONFIGURATION TIMING USING 6800 MPU FAMILY



where: m,n are integers;
T_c is character period

ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 V ±5%, V_{SS} = 0, T_A = 0 to 70°C unless otherwise noted)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage		V_{IH}	2.0	–	V_{CC}	Vdc
Input Low Voltage		V_{IL}	-0.3	–	0.8	Vdc
Input Leakage Current		I_{in}	–	1.0	2.5	μAdc
Three-State (V_{CC} = 5.25 V) (V_{in} = 0.4 to 2.4 V)		I_{TSI}	-10	2.0	10	μAdc
Output High Voltage		V_{OH}				Vdc
(I_{load} = -205 μA)	D0-D7		2.4	–	–	
(I_{load} = -100 μA)	Other Outputs		2.4	–	–	
Output Low Voltage (I_{load} = 1.6 mA)		V_{OL}	–	–	0.4	Vdc
Power Dissipation		P_D	–	600	–	mW
Input Capacitance		C_{in}				pF
	D0-D7		–	–	12.5	
	All others		–	–	10	
Output Capacitance	All Outputs	C_{out}	–	–	10	pF
Minimum Clock Pulse Width, Low		P_{WCL}	160	–	–	ns
Minimum Clock Pulse Width, High		P_{WCH}	200	–	–	ns
Clock Frequency		f_c	–	–	2.5	MHz
Rise and Fall Time for Clock Input		t_{cr}, t_{cf}	–	–	20	ns
Memory Address Delay Time		t_{MAD}	–	–	160	ns
Raster Address Delay Time		t_{RAD}	–	–	160	ns
Display Timing Delay Time		t_{DTD}	–	–	300	ns
Horizontal Sync Delay Time		t_{HSD}	–	–	300	ns
Vertical Sync Delay Time		t_{VSD}	–	–	300	ns
Cursor Display Timing Delay Time		t_{CDD}	–	–	300	ns
Light Pen Strobe Minimum Pulse Width		PW_{LPH}	100	–	–	ns
Light Pen Strobe Disable Time		t_{LPD1}	–	–	120	ns
		t_{LPD2}	–	–	0	ns

Note: The light pen strobe must fall to low level before VSYNC pulse rises.

BUS TIMING CHARACTERISTICS

Characteristic	Symbol	Min	Max	Unit
READ/WRITE				
Enable Cycle Time	t_{cycE}	1.0	–	μs
Enable Pulse Width, High	PW_{EH}	0.45	25	μs
Enable Pulse Width, Low	PW_{EL}	0.43	–	μs
Setup Time, $\overline{CS}$ and RS valid to enable positive transition	t_{AS}	160	–	ns
Data Delay Time	t_{DDR}	–	320	ns
Data Hold Time (Read)	t_H	10	–	ns
(write)		10	–	
Address Hold Time	t_{AH}	10	–	ns
Rise and Fall Time for Enable Input	t_{Er}, t_{Ef}	–	25	ns
Data Setup Time	t_{DSW}	195	–	ns
Data Access Time	t_{ACC}	–	480	ns

FIGURE 3 – CRTC TIMING CHART

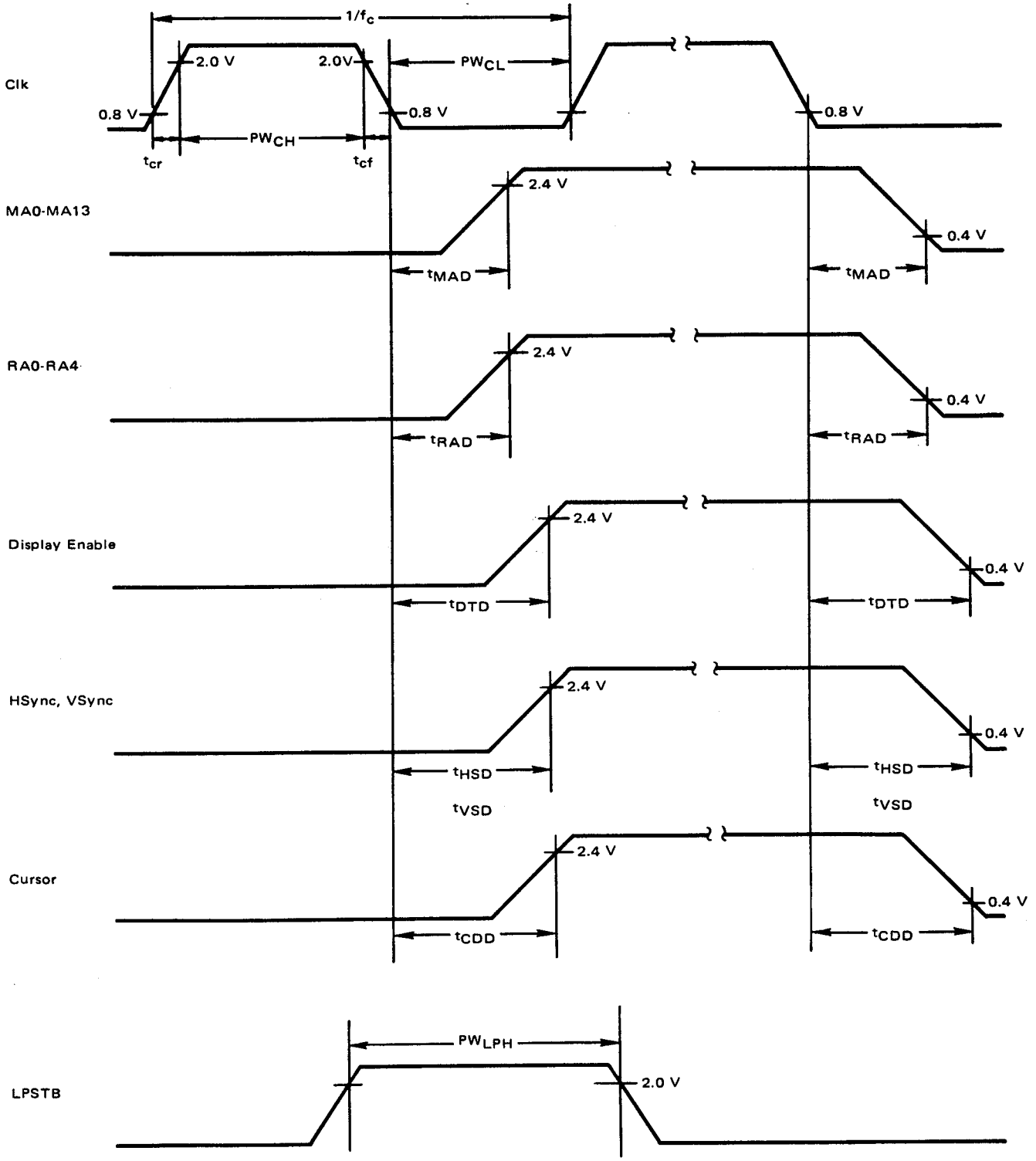


FIGURE 4 – RELATION BETWEEN LPSTB AND REFRESH MEMORY ADDRESS

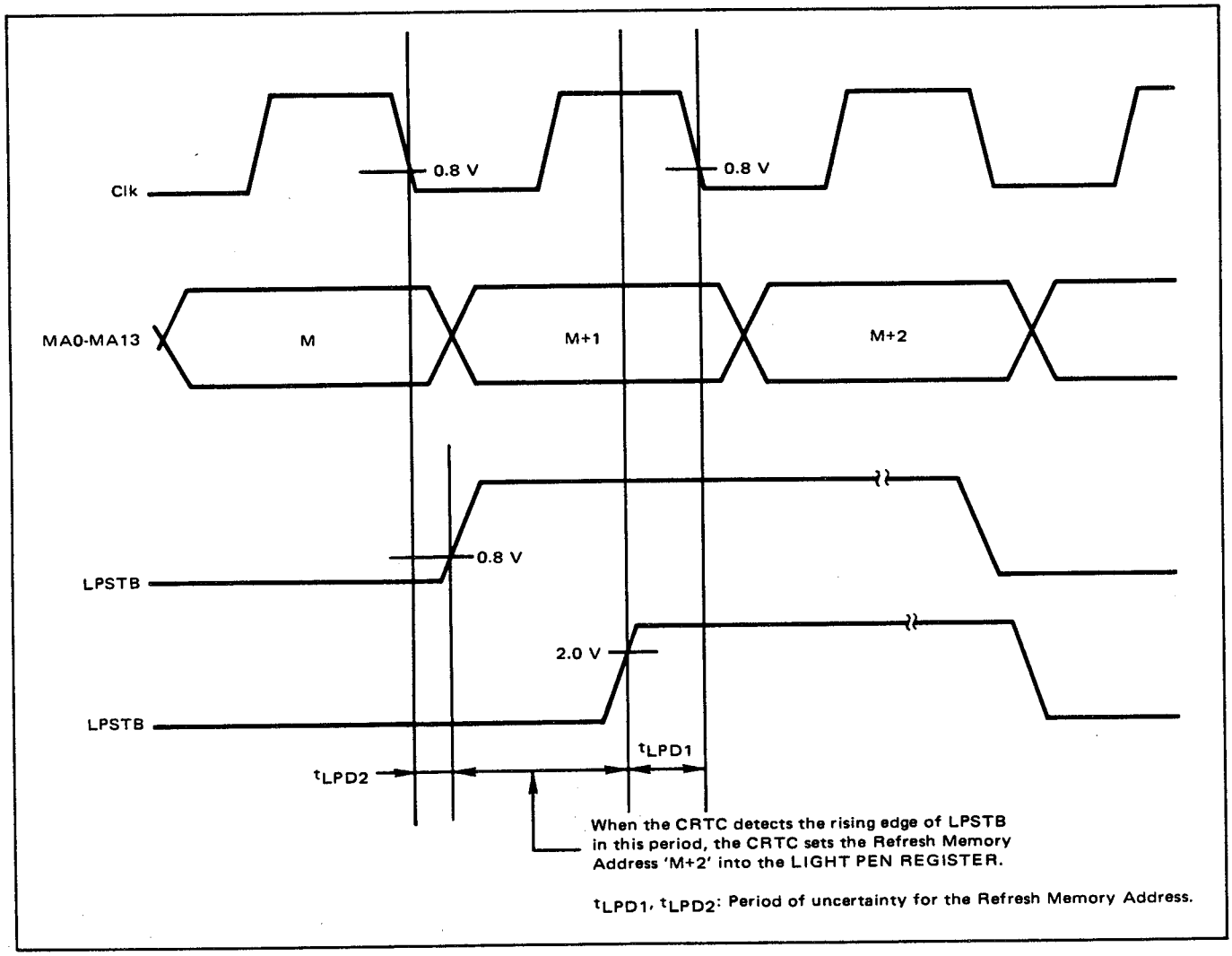
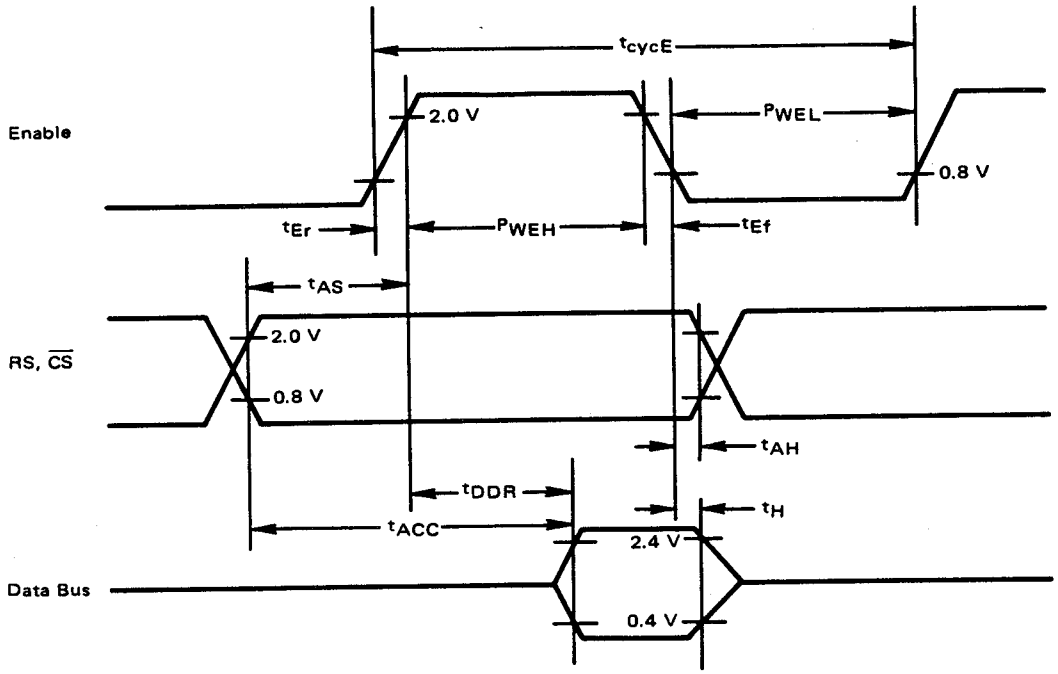


FIGURE 5 – BUS TIMING CHART

5a – Bus Read Timing (Read Information From CRTC)



5b – Bus Write Timing (Write Information Into CRTC)

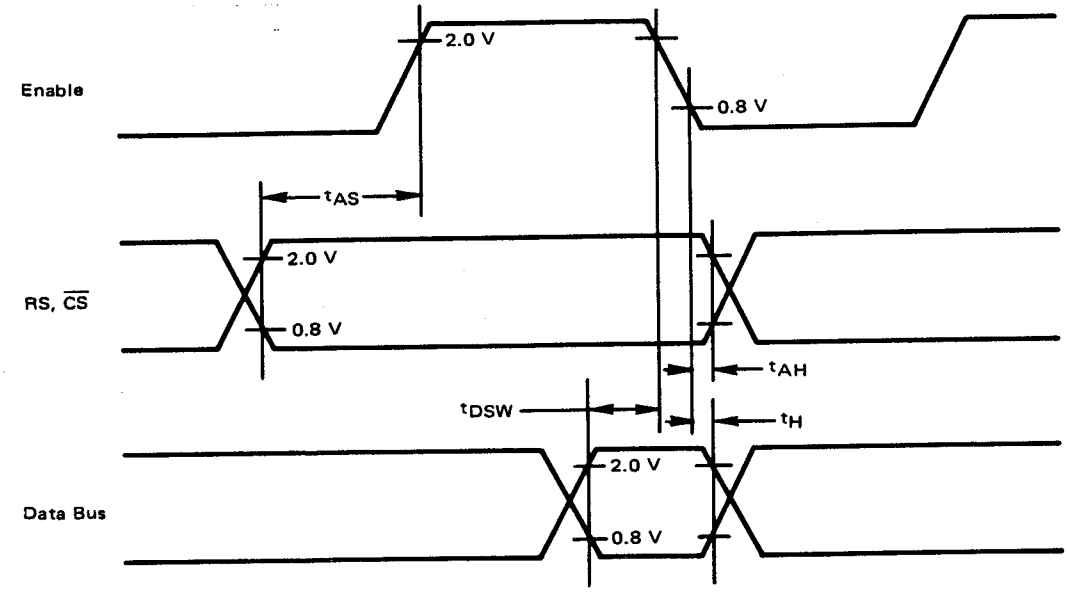


FIGURE 6 – BUS TIMING TEST LOAD

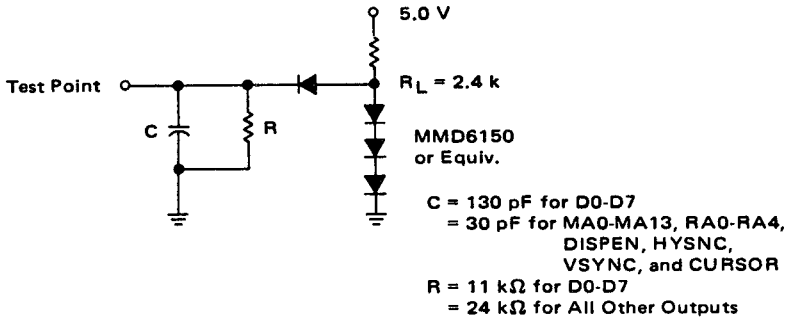
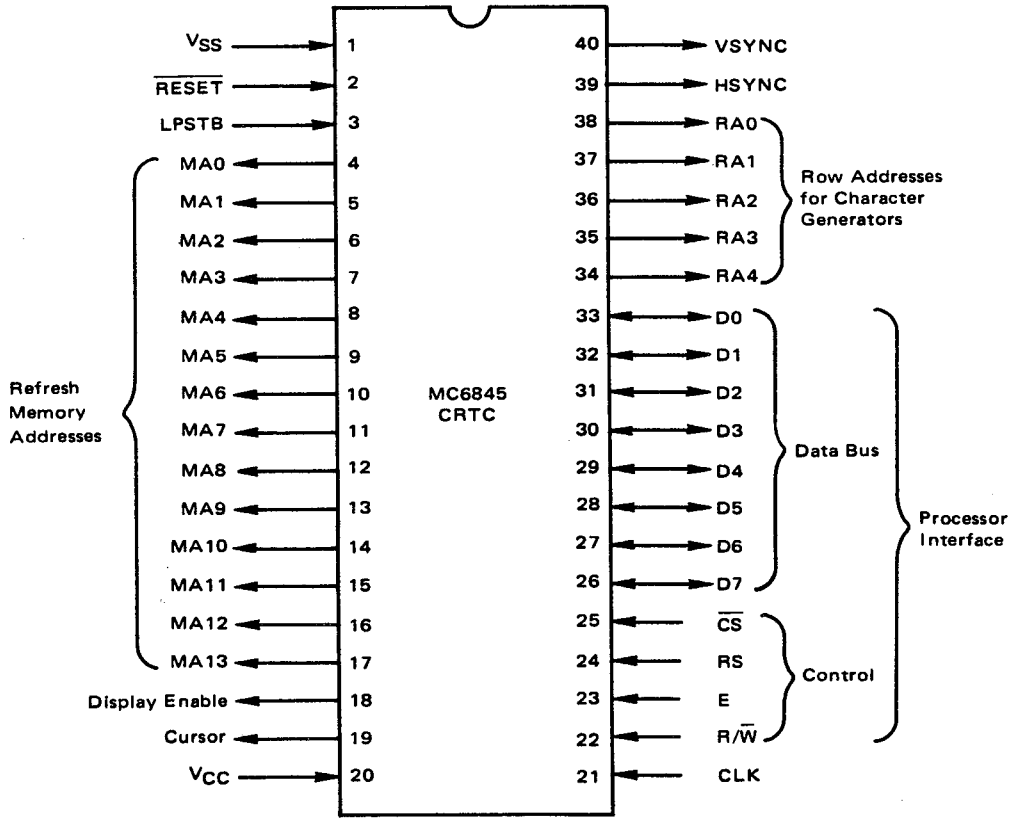


FIGURE 7 – PIN ASSIGNMENT



PIN DESCRIPTION

PROCESSOR INTERFACE

The CRTC interfaces to a processor bus on the bidirectional data bus (D0-D7) using $\overline{CS}$, RS, E, and R/$\overline{W}$ for control signals.

Data Bus (D0-D7) — The bidirectional data lines (D0-D7) allow data transfers between the CRTC internal Register File and the processor. Data bus output drivers are 3-state buffers which remain in the high impedance state except when the processor performs a CRTC read operation. A high level on a data pin is a logical "1."

Enable (E) — The Enable signal is a high impedance TTL/MOS compatible input which enables the data bus input/output buffers and clocks data to and from the CRTC. This signal is usually derived from the processor clock, and the high to low transition is the active edge.

Chip Select ($\overline{CS}$) — The $\overline{CS}$ line is a high impedance TTL/MOS compatible input which selects the CRTC when low to read or write the internal Register File. This signal should only be active when there is a valid stable address being decoded from the processor.

Register Select (RS) — The RS line is a high impedance TTL/MOS compatible input which selects either the Address Register (RS = "0") or one of the Data Registers (RS = "1") of the internal Register File.

Read/Write (R/$\overline{W}$) — The R/$\overline{W}$ line is a high impedance TTL/MOS compatible input which determines whether the internal Register File gets written or read. A write is active low ("0").

CRT CONTROL

The CRTC provides horizontal sync (HS), vertical sync (VS), and Display Enable signals.

Vertical Sync (V SYNC) — This TTL compatible output is an active high signal which drives the monitor directly or is fed to Video Processing Logic for composite generation. This signal determines the vertical position of the displayed text.

Horizontal Sync (H SYNC) — This TTL compatible output is an active high signal which drives the monitor directly or is fed to Video Processing Logic for composite generation. This signal determines the horizontal position of the displayed text.

Display Enable — This TTL compatible output is an active high signal which indicates the CRTC is providing addressing in the active Display Area.

REFRESH MEMORY/CHARACTER GENERATOR ADDRESSING

The CRTC provides Memory Addresses (MA0-MA13) to scan the Refresh RAM. Also provided are Raster Addresses (RA0-RA4) for the character ROM.

Refresh Memory Addresses (MA0-MA13) — These 14 outputs are used to refresh the CRT screen with pages of data located within a 16K block of refresh memory. These outputs drive a TTL load and 30pF. A high level on MA0-MA13 is a logical "1."

Raster Addresses (RA0-RA4) — These 5 outputs from the internal Raster Counter address the Character ROM for the row of a character. These outputs drive a TTL load and 30pF. A high level (on RA0-RA4) is a logical "1."

OTHER PINS

Cursor — This TTL compatible output indicates Cursor Display to external Video Processing Logic. Active high signal.

Clock (CLK) — The CLK TTL/MOS compatible input is used to synchronize all CRT control signals. An external dot counter is used to derive this signal which is usually the character rate in an alphanumeric CRT. The active transition is high to low.

Light Pen Strobe (LPSTR) — This high impedance TTL/MOS compatible input latches the current Refresh Addresses in the Register File. Latching is on the low to high edge and is synchronized internally to character clock.

V_{CC}, Gnd

$\overline{RES}$ — The $\overline{RES}$ input is used to Reset the CRTC. An input low level on $\overline{RES}$ forces CRTC into following status:

(A) All the counters in CRTC are cleared and the device stops the display operation.
(B) All the outputs go down to low level.
(C) Control registers in CRTC are not affected and remain unchanged.

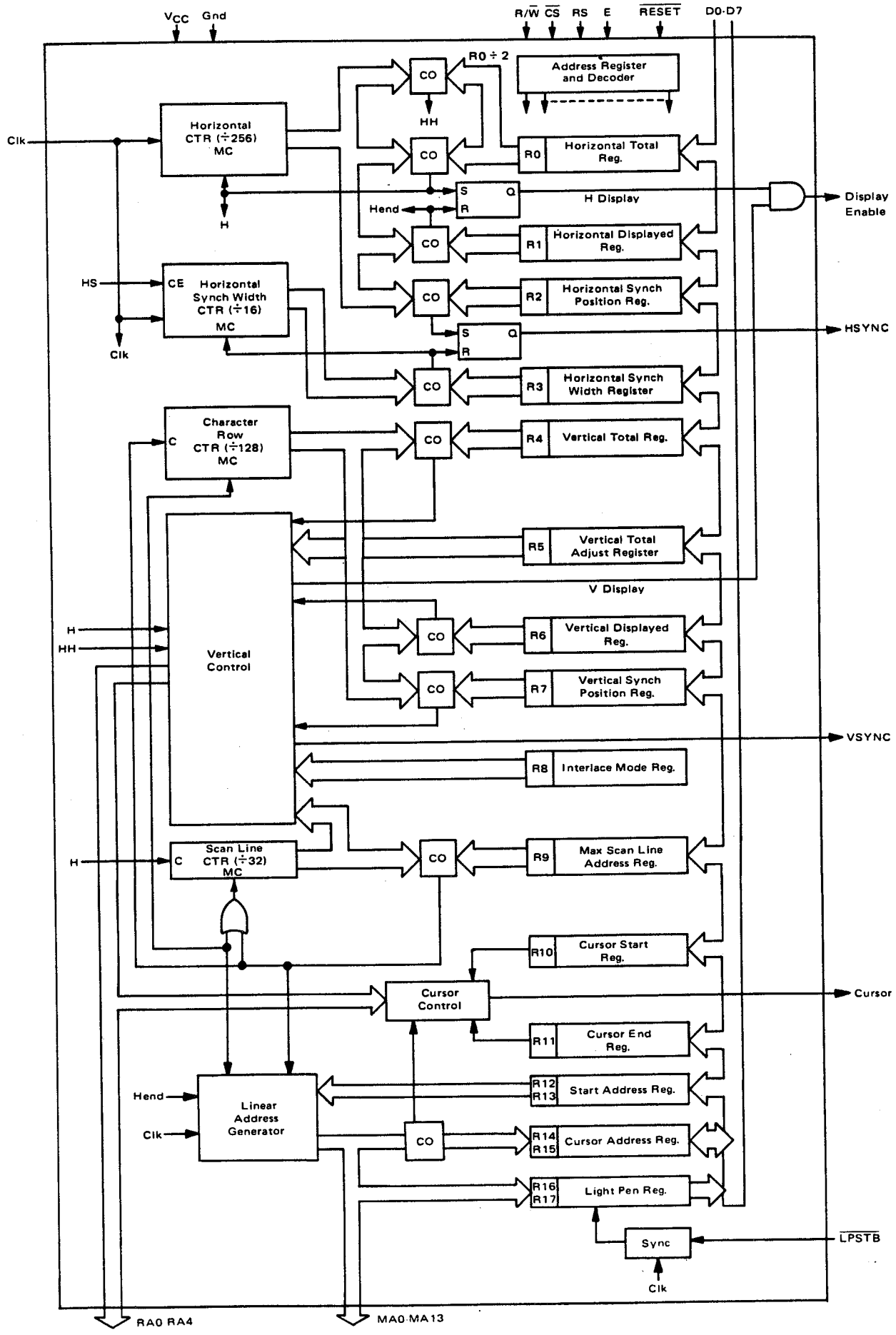
This signal is different from other M6800 family in the following functions:

(A) $\overline{RES}$ signal has capability of reset function only when LPSTB is at low level.
(B) After $\overline{RES}$ has gone down to low level, output signals of MA0-MA13 and RA0-RA4, synchronizing with CLK low level, goes down to low level. (At least 1 cycle CLK signal is necessary for reset.)
(C) The CRTC starts the Display operation immediately after the release of $\overline{RES}$ signal.

TABLE 1 — CRTC Operating Mode

$\overline{RES}$	LPSTB	OPERATING MODE
0	0	Reset
0	1	Test Mode
1	0	Normal Mode
1	1	Normal Mode

FIGURE 8 – CRTC FUNCTIONAL BLOCK DIAGRAM



CRTC DESCRIPTION
(Figure 8: CRTC Block Diagram)

The CRTC consists of programmable horizontal and vertical timing generators, programmable linear address register, programmable cursor logic, light pen capture register, and control circuitry for interface to a processor bus.

All CRTC timing is derived from CLK, usually the output of an external dot rate counter. Coincidence (CO) circuits continuously compare counter contents to the contents of the programmable register file, R0-R17. For horizontal timing generation, comparisons result in: 1) Horizontal sync pulse (HS) of a frequency, position, and width determined by the registers, 2) Horizontal Display Signal of a frequency, position, and duration determined by the registers.

The Horizontal counter produces H clock which drives the Scan Line Counter and Vertical Control. The contents of the Raster Counter are continuously compared to the Max Scan Line Address Register. A coincidence resets the Raster Counter and clocks the Vertical Counter.

Comparisons of Vertical Counter contents and Vertical Registers result in: 1) Vertical sync pulse (VS) of a frequency and positon determined by the registers—the width is fixed at 16 raster lines in the vertical control section and is not programmable, 2) Vertical Display of a frequency and position determined by the registers.

The Vertical Control Logic has other functions.

1. Generate row selects, RA0-RA4, from the Raste Count for the corresponding interlace or non interlace modes.
2. Extend the number of scan lines in the vertica total by the amount programmed in the Vertica Total Adjust Register.

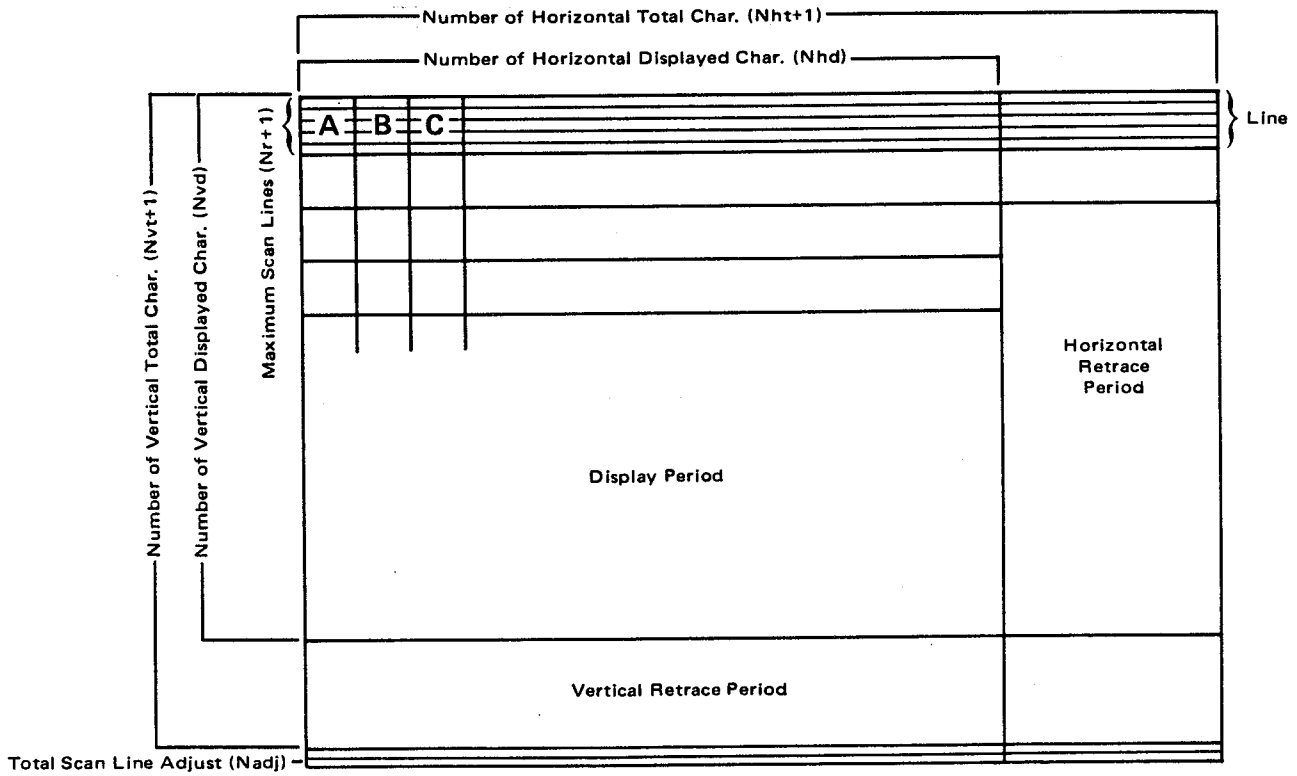
The Linear Address Generator is driven by CLK and locates the relative positions of characters in memory with their positions on the screen. Fourteen lines MA0-MA13, are available for addressing up to fou pages of 4K characters, 8 pages of 2K characters, etc Using the Start Address Register, hardware scrollin through 16K characters is possible. The Linear Addres Generator repeats the same sequence of addresses for eacl scan line of a character row.

The cursor logic determines the cursor location, size and blinking rate on the screen. All are programmable.

The light pen strobe going high causes the curren contents of the Address Counter to be latched in th Light Pen Register. The contents of the Light Pen Registe are subsequently read by the Processor.

Internal CRTC registers are programmed by th processor through the data bus, D0-D7, and the contro signals—R/$\overline{W}$, $\overline{CS}$, RS and E.

FIGURE 9 – ILLUSTRATION OF THE CRT SCREEN FORMAT



REGISTER FILE DESCRIPTION
(See Table 2)

Nineteen registers in the CRTC can be accessed by means of the data bus. Register addressing and lengths are shown in Table 2.

ADDRESS REGISTER

The Address Register is a 5 bit write-only register used as an "indirect" or "pointer" register. Its contents are the address of one of the other 18 registers in the file. When RS and $\overline{CS}$ are low, the Address Register itself is addressed. When RS is high, the Register File is accessed.

HORIZONTAL TIMING REGISTERS R0, R1, R2, and R3

Figure 9 shows the visible display area of a typical CRT monitor giving the point of reference for horizontal registers as the left most displayed character position. Horizontal registers are programmed in "character time" units with respect to the reference.

Horizontal Total Register (R0) – This 8 bit write-only register determines the horizontal frequency of HS. It is the total of displayed plus non-displayed character time units minus one.

Horizontal Displayed Register (R1) – This 8 bit write-only register determines the number of displayed characters per horizontal line.

Horizontal Sync Position Register (R2) – This 8 bit write-only register determines the horizontal sync postiion on the horizontal line.

Horizontal Sync Width Register (R3) – This 4 bit write-only register determines the width of the HS pulse. It may not be apparent why this width needs to be programmed. However, consider that all timing widths must be programmed as multiples of the character clock period which varies. If HS width were fixed as an integral number of character times, it would vary with character rate and be out of tolerance for certain monitors. The rate programmable feature allows compensating HS width.

VERTICAL TIMING REGISTERS R4, R5, R6, R7, R8, and R9

The point of reference for vertical registers is the top character position displayed. Vertical registers are programmed in character row times or scan line times.

Vertical Total Register (R4) and Vertical Total Adjust Register (R5) – The vertical frequency of VS is determined by both R4 and R5. The calculated number of character line times is usually an integer plus a fraction to get exactly a 50 or 60 Hz vertical refresh rate. The integer number of character line times minus one is programmed in the 7 bit write-only Vertical Total Register; the fraction is programmed in the 5 bit write-only Vertical Scan Adjust Register as a number of scan line times.

Vertical Displayed Register (R6) – This 7 bit write-only register determines the number of displayed character rows on the CRT screen, and is programmed in character row times.

Vertical Sync Position (R7) – This 7 bit write-only register determines the vertical sync position with respect

TABLE 2 – CRTC INTERNAL REGISTER ASSIGNMENT

$\overline{CS}$	RS	Address Register 4	3	2	1	0	Register #	Register File	Program Unit	Read	Write	Number of Bits 7	6	5	4	3	2	1	0
1	X	X	X	X	X	X	X	–	–	–	–								
0	0	X	X	X	X	X	X	Address Register	–	No	Yes								
0	1	0	0	0	0	0	R0	Horizontal Total	Char.	No	Yes								
0	1	0	0	0	0	1	R1	Horizontal Displayed	Char.	No	Yes								
0	1	0	0	0	1	0	R2	H. Sync Position	Char.	No	Yes								
0	1	0	0	0	1	1	R3	H. Sync Width	Char.	No	Yes								
0	1	0	0	1	0	0	R4	Vertical Total	Char. Row	No	Yes								
0	1	0	0	1	0	1	R5	V. Total Adjust	Scan Line	No	Yes								
0	1	0	0	1	1	0	R6	Vertical Displayed	Char. Row	No	Yes								
0	1	0	0	1	1	1	R7	V. Sync Position	Char. Row	No	Yes								
W 0	1	0	1	0	0	0	R8	Interlace Mode	–	No	Yes								
0	1	0	1	0	0	1	R9	Max Scan Line Address	Scan Line	No	Yes								
0	1	0	1	0	1	0	R10	Cursor Start	Scan Line	No	Yes		B	P			(Note 1)		
0	1	0	1	0	1	1	R11	Cursor End	Scan Line	No	Yes								
0	1	0	1	1	0	0	R12	Start Address (H)	–	No	Yes								
0	1	0	1	1	0	1	R13	Start Address (L)	–	No	Yes								
0	1	0	1	1	1	0	R14	Cursor (H)	–	Yes	Yes								
0	1	0	1	1	1	1	R15	Cursor (L)	–	Yes	Yes								
0	1	1	0	0	0	0	R16	Light Pen (H)	–	Yes	No								
0	1	1	0	0	0	1	R17	Light Pen (L)	–	Yes	No								

Note (1): Bit 5 of the Cursor Start Raster Register is used for blink period control, and Bit 6 is used to select blink or non-blink.

to the reference. It is programmed in character row times.

Interlace Mode Register (R8) — This 2 bit write-only register controls the raster scan mode (see Figure 11). When bit 0 and bit 1 are reset, or bit 0 is reset and bit 1 set, the non-interlace raster scan mode is selected. Two interlace modes are available. Both are interlaced 2 fields per frame. When bit 0 is set and bit 1 is reset, the interlace sync raster scan mode is selected. Also when bit 0 and bit 1 are set, the interlace sync and video raster scan mode is selected.

Maximum Scan Line Address Register (R9) — This 5 bit write-only register determines the number of scan lines per character row including spacing. The programmed value is a max address and is one less than the number of scan lines.

OTHER REGISTERS

Cursor Start Register (R10) — This 7 bit write-only register controls the cursor format (see Figure 10). Bit 5 is the blink timing control. When bit 5 is low, the blink frequency is 1/16 of the vertical field rate, and when bit 5 is high, the blink frequency is 1/32 of the vertical field rate. Bit 6 is used to enable a blink. The cursor start scan line is set by the lower 5 bits.

Cursor End Register (R11) — This 5 bit write-only register sets the cursor end scan line.

Start Address Register (H & L) (R12, R13) — Start Address Register is a 14 bit write-only register which determines the first address put out as a refresh address after vertical blanking. It consists of an 8 bit lower register, and a 6 bit higher register.

Light Pen Register (H & L) (R16, R17) — This 14 bit read-only register is used to store the contents of the Address Register (H & L) when the LPSTB input pulses high. This register consists of an 8 bit lower and 6 bit higher register.

Cursor Register (H & L) (R14, R15) — This 14 bit read/write register stores the cursor location. This register consists of an 8 bit lower and 6 bit higher register.

CURSOR

The Cursor Start and End Registers allow a cursor of up to 32 scan lines in height to be placed on any scan lines of the character block as shown in Figure 10. Using Bits 5 & 6 of the Cursor Start Register, the cursor is programmed with blink periods of 16 or 32 times the field period. Optional non-blink and non-display modes can also be selected. When an external 2X blink on characters is required, it may be necessary to perform cursor blink externally as well so that both blink rates are synchronized. Note that an invert/non-invert cursor is easily implemented by programming the CRTC for blinking cursor and externally inverting the video signal with an exclusive-OR.

The cursor is positioned by changing the contents of registers R14 and R15. The cursor can be placed at any of 16K character positions, thus facilitating hardware paging and scrolling through memory without loss of the cursor's original position.

INTERLACE/NON-INTERLACE DISPLAY MODES

An illustration of the 3 raster scan modes of operation is shown in Figure 11. Normal sync mode is non-interlace. In this mode, each scan line is refreshed at the vertical field rate (e.g., 50 or 60 Hz). Frame time is divided into even and odd alternating fields. The horizontal and vertical timing relationship results in the displacement of scan lines in the odd field with respect to the even field. When the same information is painted in both fields, the mode is called "Interlace Sync;" this is a useful mode for enhancing readability by filling in a character. When the even lines of a character are displayed in the even field and the odd lines in the odd field, the mode is called "Interlace Sync and Video." This last mode effectively doubles the character density on a monitor of a given bandwidth. The disadvantage of both interlace modes is an apparent flicker effect, which can be reduced by careful monitor design.

There are restrictions on the programming of CRTC registers for interlace operation:

1) Horizontal total character count, N_{ht} must be odd (i.e., an even number of character times)
2) For Interlace Sync and Video mode only, the max scan line address, N_{sl}, must be odd (i.e., an even number of scan lines)
3) For Interlace Sync and Video mode only, the Vertical Displayed Total characters must be even. The programmed number, N_{vd}, must be *one-half* the actual number required.
4) For Interlace Sync & Video mode only, the Cursor START and Cursor End Registers must both be even or both odd.

LIGHT PEN

The contents of the CRTC Address Counter are strobed into R16/R17 Light Pen Registers on the next high to low CLK transition after LPSTB goes high. In most systems, the light pen signal would also cause a processor interrupt routine to read R16/R17. Slow light pen response requires the processor software to modify the captured address read from R16/R17 by a calibration factor.

PROGRAMMING CONSIDERATIONS

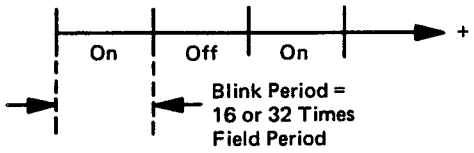
Initialization — Registers R0-R15 must be initialized after power is turned on. The processor normally loads the CRTC registers sequentially from a firmware table. Henceforth, R0-R11 are not changed in most systems. The 6800 program in Table 3 and Figure 12 shows a typical CRTC initialization.

Hardware Scrolling — Registers R12/R13 contents determine which memory location is the first displayed character on the screen. Since the CRTC Linear Address Generator counts from this beginning count, the displayed portion of the screen may be a window on any continuous string of characters within a 16K block or refresh memory. By centering the R12/R13 pointer in the middle of the available memory space, scrolling up or down is possible . . . by line, page, or character.

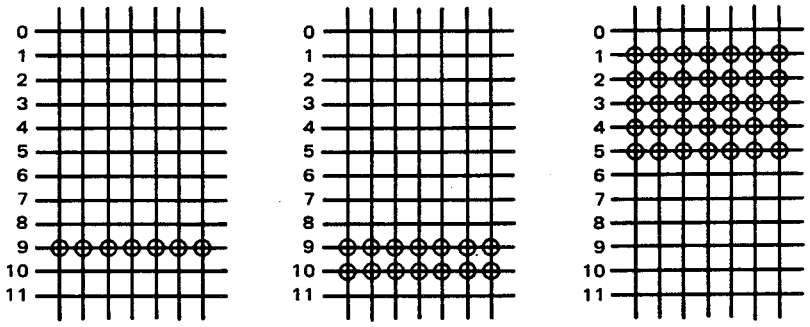
FIGURE 10 – CURSOR CONTROL

Cursor Start Register

B Bit 6	P Bit 5	Cursor Display Mode
0	0	Non-Blink
0	1	Cursor Non-Display
1	0	Blink, 1/16 Field Rate
1	1	Blink, 1/32 Field Rate



Example of Cursor Display Mode



Cursor Start Adr. = 9
Cursor End Adr. = 9

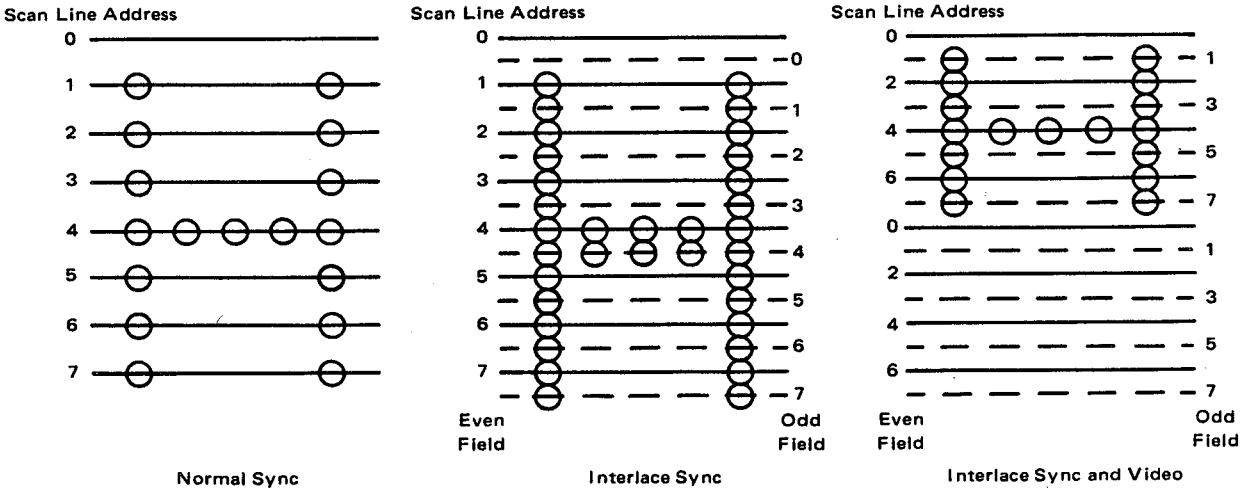
Cursor Start Adr. = 9
Cursor End Adr. = 10

Cursor Start Adr. = 1
Cursor End Adr. = 5

FIGURE 11 – INTERFACE CONTROL

Interlace Mode Register

Bit 1	Bit 0	Mode
0 1	0 0	Normal Sync Mode (Non-Interlace)
0	1	Interlace Sync Mode
1	1	Interlace Sync & Video Mode



Normal Sync

Interlace Sync

Interlace Sync and Video

TABLE 3 — Typical 80 x 24 Screen Format Initialization of CRTC

Reg. #	Register File	Program Unit	Calculation*	Programmed Value	
				Decimal	Hex
R0	H Total	T_c	102 x .527 = 53.76 μs	102 - 1 = 101	N_{ht} = $65
R1	H Displayed	T_c	80 x .527 = 42.16 μs	80	N_{hd} = $50
R2	H Sync Position	T_c	86 x .527 = 45.32 μs	86	N_{hsp} = $56
R3	H Sync Width	T_c	9 x .527 = 4.74 μs	9	N_{hsw} = $09
R4	V Total	T_{cr}	25 x 645.12 = 16.13 ms	25 = 1 = 24	N_{vt} = $18
R5	V Total Adjust	T_{sl}	10 x 53.76 = .54 ms	10	N_{adj} = $0A
R6	V Displayed	T_{cr}	24 x 645.12 = 15.48 ms	24	N_{vd} = $18
R7	V Sync Position	T_{cr}	24 x 645.12 = 15.48 ms	24	N_{vsp} = $18
R8	Interlace Mode	—		—	$00
R9	Max Scan Line Address	T_{sl}		11	N_{sl} = $0B
R10	Cursor Start	T_{sl}		0	$00
R11	Cursor End	T_{sl}		11	$0B
R12	Start Address (H)	—		128	$00
R13	Start Address (L)	—			$80
R14	Cursor (H)	—		128	$00
R15	Cursor (L)	—			$80

Clock Period = T_c = .527 μs

Scan Line Period = T_{sl} = $(N_{ht} + 1) \times T_c$ = 102 x .527 μs = 53.76 μs

Character Row Period = $T_{cr} = N_{sl} \times T_{sl}$ = 12 x 53.76 μs = 645.12 μs

*These are typical values for the Motorola M3000 Monitor; values may vary for other monitors.

FIGURE 12 — INITIALIZATION OF CRTC FOR 80x24 SCREEN FORMAT IN TABLE 3

```
PAGE  001   CRTINT

00001                          NAM     CRTINT
00002 0000                     ORG     $0
00003 0000 5F                  CLR B           CLEAR COUNTER
00004 0001 CE 0020             LDX     #$20
00005 0004 F7 9000 CRTII       STA B   $9000   CRTC ADDR REG
00006 0007 A6 00               LDA A   0,X
00007 0009 B7 9001             STA A   $9001   ACC TO CRTC REG
00008 000C 08                  INX
00009 000D 5C                  INC B           INC COUNTER
00010 000E C1 10               CMP B   #$10    LAST CRTC REG?
00011 0010 26 F2               BNE     CRTII
00012 0012 3F                  SWI
00013 0020                     ORG     $20
00014 0020 65          CRTTAB  FCB     $65,$50,$56,$9
00015 0024 18                  FCB     $18,$0A,$18,$18
00016 0028 00                  FCB     0,$0B,0,$0B
00017 002C 0080                FDB     $80,$80
00018      0000                END
CRTII  0004 CRTTAB 0020

TOTAL ERRORS 00000
```

OPERATION OF THE CRTC

Timing Chart of the CRT Interface Signals – Timing charts of CRT interface signals are illustrated in this section with the aid of programmed example of the CRTC. When values listed in Table 4 are programmed into CRTC control registers, the device provides the outputs as shown in the Timing Diagrams (Figures 13 through 15). The screen format of this example is shown in Figure 9. Figure 16 is an illustration of the relation between Refresh Memory Address (MA0-MA13), Raster Address (RA0-RA4) and the position on the screen. In this example, the start address is assumed to be "0".

TABLE 4 – Values Programmed Into CRTC Registers

Reg. #	Register Name	Value	Programmed Value
R0	H. Total	$N_{ht}+1$	N_{ht}
R1	H. Displayed	N_{hd}	N_{hd}
R2	H. Sync Position	N_{hsp}	N_{hsp}
R3	H. Sync Width	N_{hsw}	N_{hsw}
R4	V. Total	$N_{vt}+1$	N_{vt}
R5	V. Scan Line Adjust	N_{adj}	N_{adj}
R6	V. Displayed	N_{vd}	N_{vd}
R7	V. Sync Position	N_{vsp}	N_{vsp}
R8	Interlace Mode		
R9	Max. Scan Line Address	N_{sl}	N_{sl}
R10	Cursor Start		
R11	Cursor End		
R12	Start Address (H)	0	
R13	Start Address (L)	0	
R14	Cursor (H)		
R15	Cursor (L)		
R16	Light Pen (H)		
R17	Light Pen (L)		

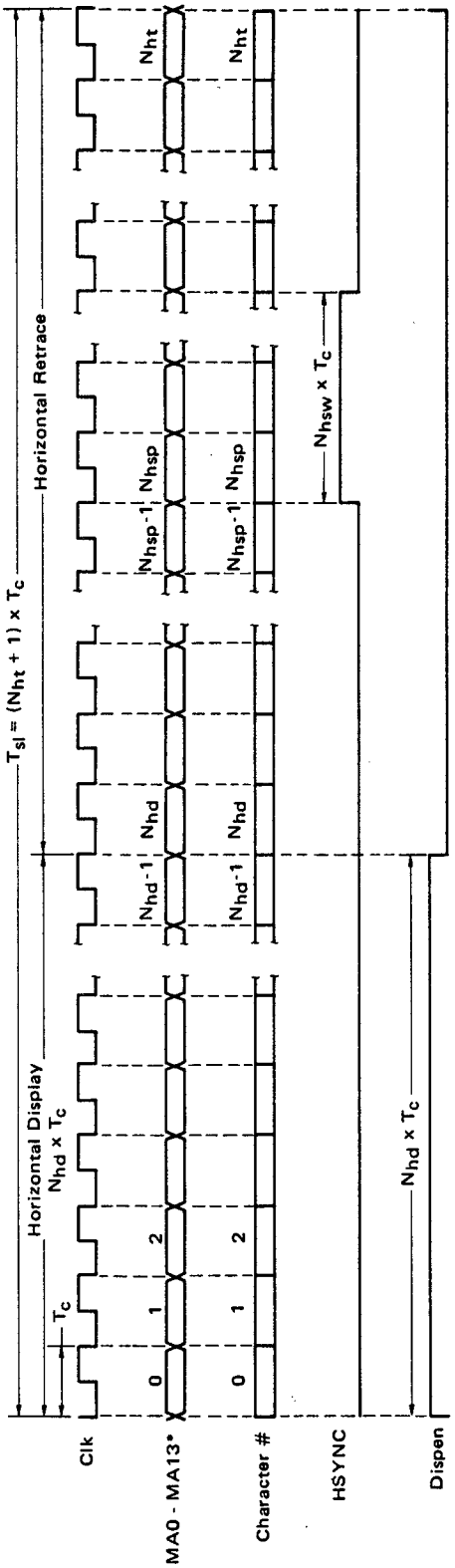
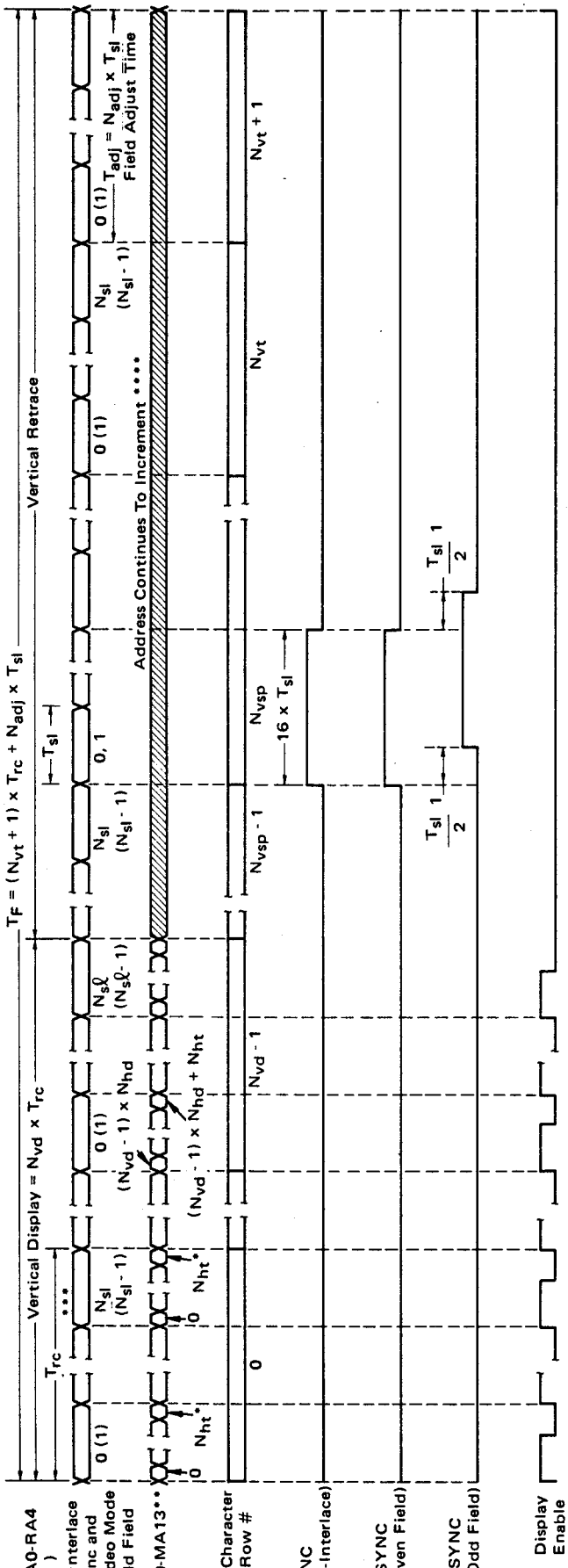


FIGURE 13 – CRTC HORIZONTAL TIMING

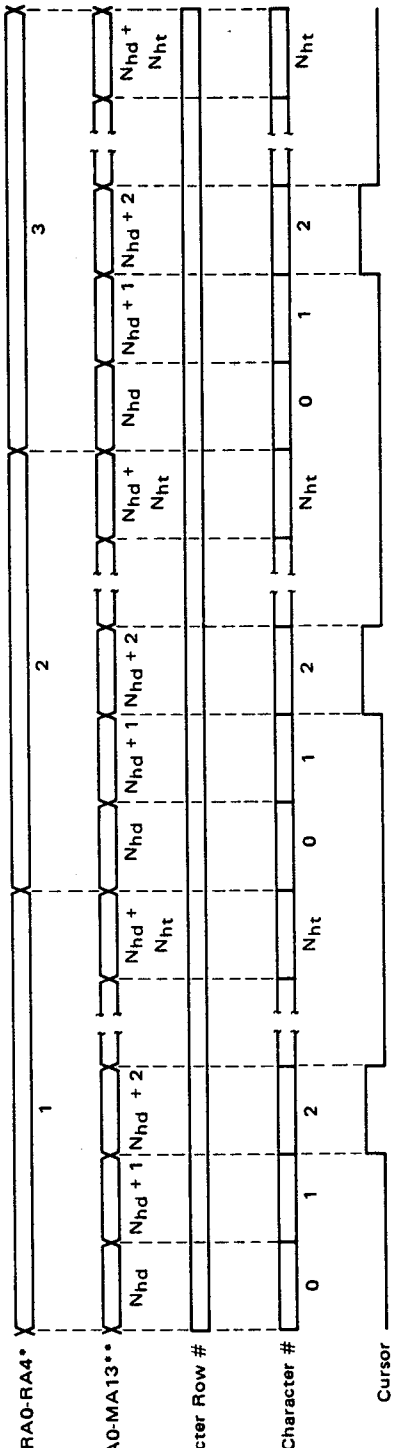
*Timing is shown for first displayed scan row only. See Chart in Figure 16 for other rows. The initial MA is determined by the contents of Start Address Register, R12/R13. Timing is shown for R12/R13 = 0.

FIGURE 14 – CRTC VERTICAL TIMING



*N_{ht} must be an odd number for both interlace modes.
**Initial MA is determined by R12/R13 (Start Address Register), which is zero in this timing example.
***N_{sl} must be an odd number for Interlace Sync and Video Mode.
****The present CRTC freezes MA addresses at $N_{vd} \times N_{hd}$ during vertical retrace. A design change is pending to allow MA to free run during vertical retrace time.

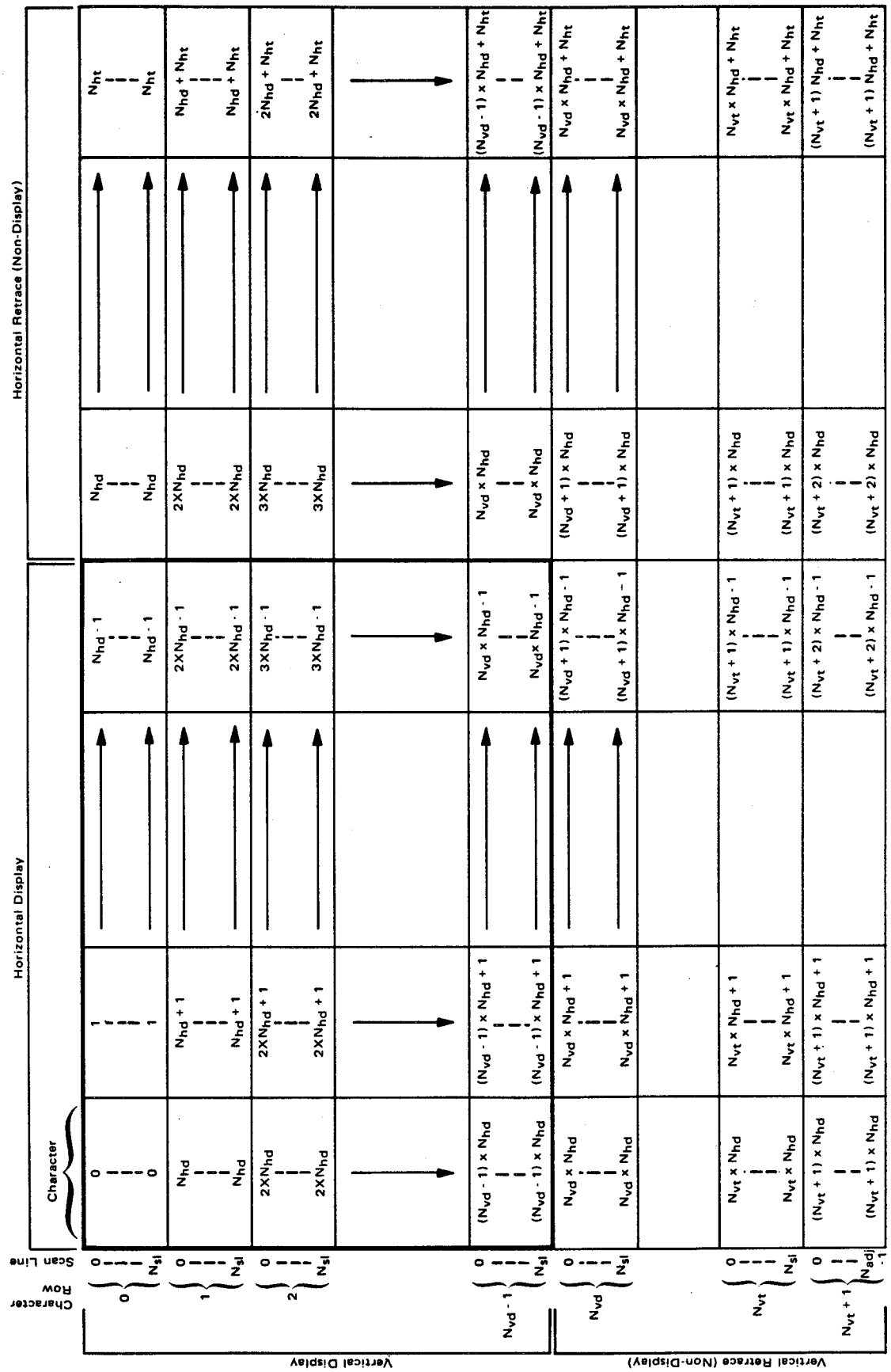
FIGURE 15 – CURSOR TIMING



*Timing is shown for non-interlace and interlace sync modes.
Example shown has cursor programmed as:
Cursor Register = $N_{hd}+2$
Cursor Start = 1
Cursor End = 3

**The initial MA is determined by the contents of Start Address Register, R12/R13. Timing is shown for R12/R13 = 0.

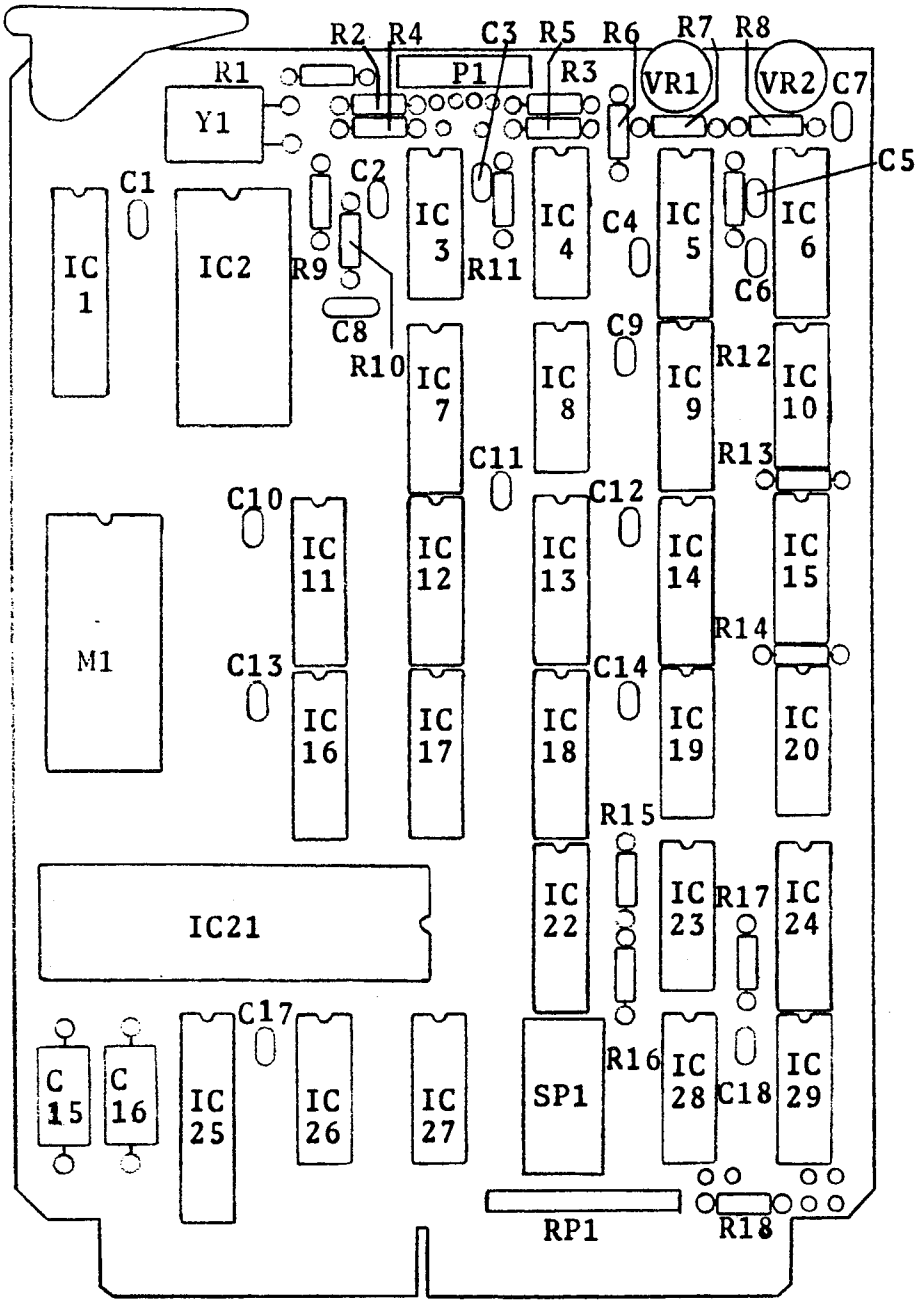
FIGURE 16 — REFRESH MEMORY ADDRESSING (MA0-MA13) STATE CHART

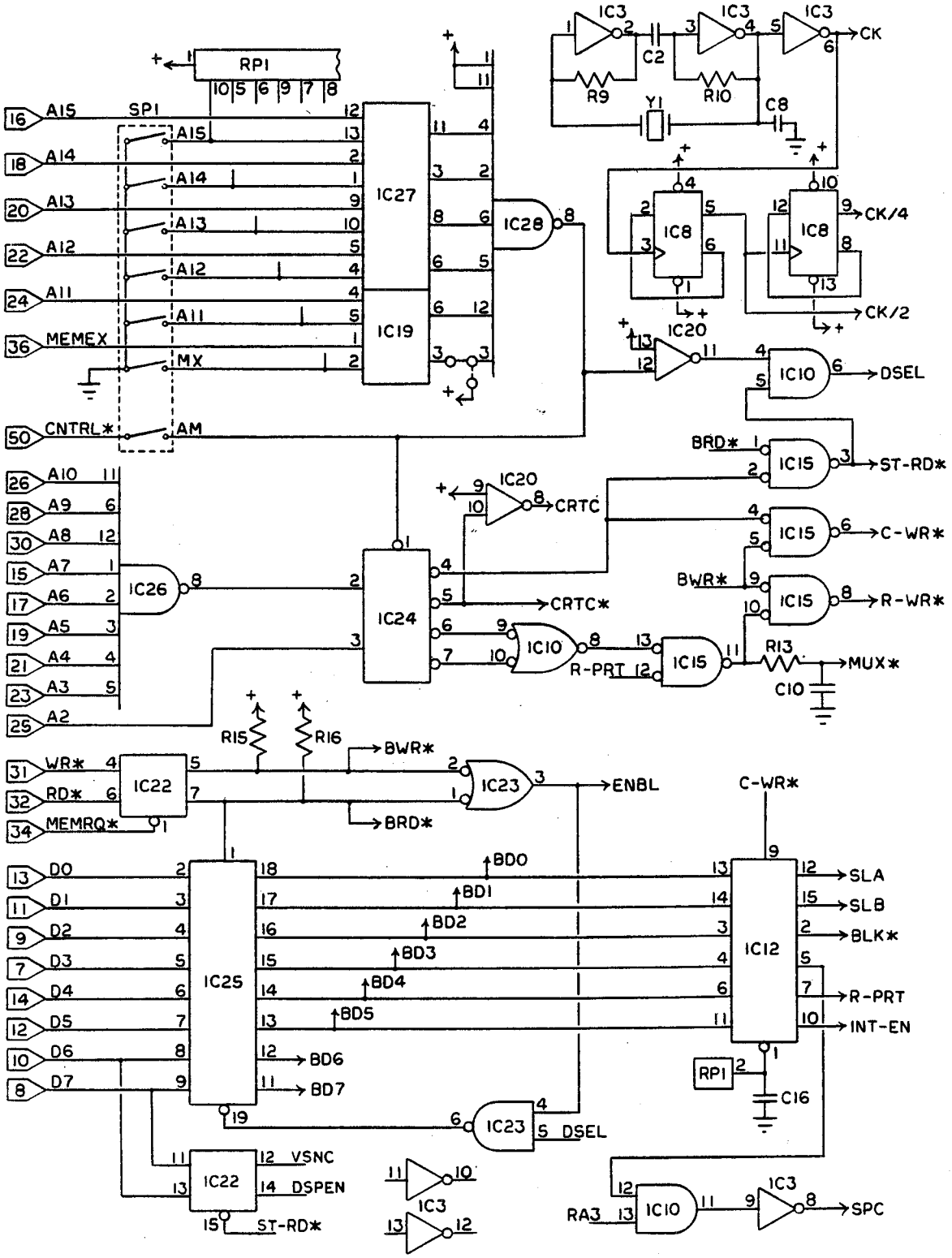


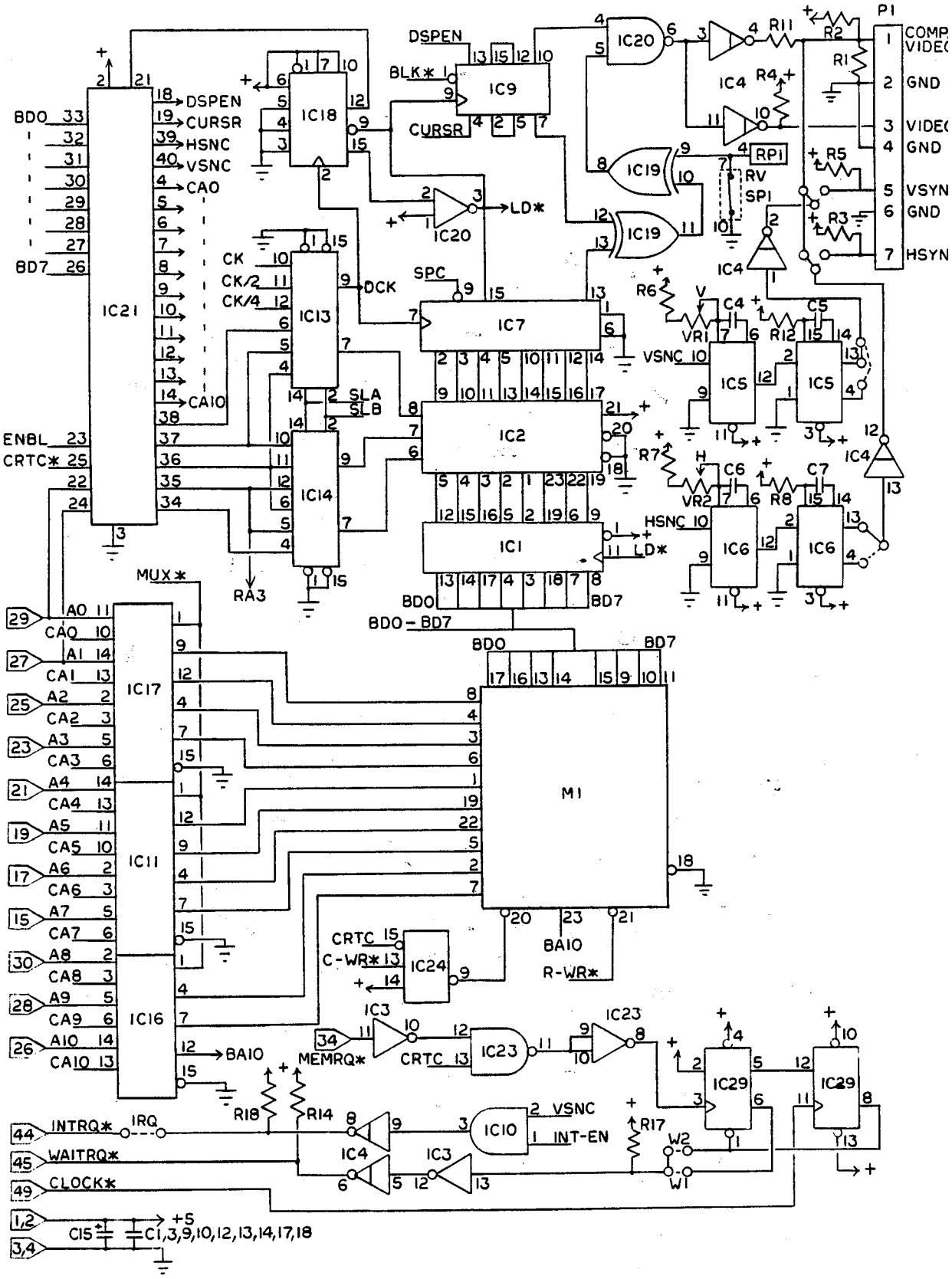
NOTE 1 The initial MA is determined by the contents of start address register, R12/R13. Timing is shown for R12/R13 = 0. Only Non-Interlace and Interlace Sync Modes are shown.

NOTE 2 The present CRTC freezes MA addresses at $N_{vd} \times N_{hd}$ during vertical retrace. A design change is pending to allow MA to free run during vertical retrace time.

STD-VID 64/80 Parts Placement Diagram 12-1-81




IC3
IC3
IC3
CK
C2
R9
R10
Y1
C8
RP1
SP1
A15
A14
A13
A12
A11
MEMEX
MX
CNTRL*
AM
IC27
IC19
IC28
IC8
IC8
CK/4
CK/2
IC20
IC10
DSEL
BRD*
IC15
ST-RD*
IC20
CRTC
IC15
C-WR*
BWR*
IC15
R-WR*
A10
A9
A8
A7
A6
A5
A4
A3
A2
IC26
IC24
CRTC*
IC10
R-PRT
IC15
R13
MUX*
C10
R15
R16
BWR*
WR*
RD*
MEMRQ*
IC22
IC23
ENBL
BRD*
C-WR*
D0
D1
D2
D3
D4
D5
D6
D7
IC25
BD0
BD1
BD2
BD3
BD4
BD5
BD6
BD7
IC12
SLA
SLB
BLK*
R-PRT
INT-EN
RP1
C16
IC23
DSEL
IC22
VSNC
DSPEN
ST-RD*
IC3
IC10
RA3
IC3
SPC


P1
COMP VIDEO
GND
VIDEO
GND
VSYN
GND
HSYN
IC21
IC18
IC9
IC20
IC19
IC4
IC13
IC14
IC7
IC2
IC1
IC5
IC6
IC17
IC11
IC16
M1
IC24
IC3
IC23
IC29
IC10
DSPEN
CURSR
HSNC
VSNC
CAO
CA10
LD*
SPC
DCK
SLA
SLB
RA3
MUX*
BD0 - BD7
BA10
R-WR*
CRTC
C-WR*
MEMRQ*
INTRQ*
IRQ
WAITRQ*
CLOCK*
INT-EN
ENBL
CRTC*
+5
C15
C1,3,9,10,12,13,14,17,18

Capacitors

C1, C3, C4, C5, C9, C10, C12, C13, C14, C17, C18	.1 mf Mono-Ceramic
C2, C6, C7, C11	680 pf 5% NPO Ceramic
C15, C16	22 mf 16V Electrolytic
C8	24 pf Mica

Crystal

Y1	12.000 MHz (64 char. version) 14.000 MHz (80 char. version)

Integrated Circuits

IC1	LS273
IC2	FP VID1
IC3	LS04
IC4	7406
IC5, IC6	LS221
IC7	LS166
IC8, IC29	LS74
IC9	LS175
IC10	LS08
IC11, IC16, IC17	LS157
IC12	LS174
IC13, IC14	LS153
IC15	LS32
IC18	LS163 or LS161
IC19, IC27	LS86

IC20, IC23	LS00
IC21	MC6845
IC22	LS367
IC24	LS139
IC25	LS245
IC26, IC28	LS30
Memory Chip	
M1	6116 type, 200 ns
Resistors	
R1, R2	120 ohm, 5%, 1/4 W
R3, R4, R5	200 ohm, 5%, 1/4 W
R6, R7	2K2 ohm, 5%, 1/4 W
R8	12K ohm, 5%, 1/4 W
R9, R10, R15, R16	330 ohm, 5%, 1/4 W
R11, R13	27 ohm, 5%, 1/4 W
R12	5K6 ohm, 5%, 1/4 W
R14, R17, R18	10K ohm, 5%, 1/4 W
Resistor Packs	
RP1	10-pin SIP, 9 Res., 4K7 ohm
Switch Pack	
SP1	8 Pos. DIP switch
Variable Resistors	
VR1	50K ohm, linear
VR2	20K ohm, linear

Miscellaneous

P1	2-pin, .1" Right angle header
J1	2-pin housing with crimp terminals
PCB	Printed circuit board
Sockets	24-pin DIP
Socket	40-pin DIP

Hardware

Card ejector	
Shipping bag	Anti-static

```
                ;*************************************************
                ;***                                           ***
                ;***  8085/Z80 Video Driver Routine            ***
                ;***  for the STD VID64/80 board               ***
                ;***                                           ***
                ;***  80 character version 1.1                 ***
                ;***  COPYRIGHT 1984 VersaLogic                ***
                ;***                                           ***
                ;*************************************************
                ;
                ;VDR ENTRY POINT=F000H, ASCII CHARACTER IN C REG
                ;VDR INIT ENTRY POINT=F003H
                ;
                ;STD VID64/80 BOARD ADDRESSED AT LOC E800H
                ;RAM WORK AREA AT 0043H-0046H, 4 BYTES
                ;
                ;
FFFF =          YES     EQU     OFFFFH  ;SET VALUES FOR IF STATEMENTS
0000 =          NO      EQU     00000H  ;
                ;
FFFF =          V80CHAR EQU     YES     ;80 CHARACTER BOARD?
0000 =          V64CHAR EQU     NO      ;64 CHARACTER BOARD?
                ;
                        IF      V80CHAR
0050 =          VSIZE   EQU     80      ;SET VSIZE FOR 80 CHAR BOARD
                        ENDIF
                ;
                        IF      V64CHAR
                VSIZE   EQU     64      ;SET VSIZE FOR 64 CHAR BOARD
                        ENDIF
                ;
E800 =          VBASE   EQU     0E800H      ;START OF STD VID 64/80 BOARD
EF7F =          VIDEND  EQU     VBASE+(24*VSIZE)-1 ;END OF DISPLAYED VID RAM
EFF8 =          VCTRL   EQU     VBASE+7F8H ;VID CONTROL AND STATUS ADD
EFFC =          CRTCSL  EQU     VBASE+7FCH ;CRTC REG SELECT ADD
EFFE =          CRTCWR  EQU     VBASE+7FEH ;CRTC WRITE PORT ADD
                ;
007F =          CURCH   EQU     7FH     ;CURSOR CHARACTER
                ;
0043 =          VCADL   EQU     043H    ;VID CUR LOC LO BYTE
0044 =          VCADH   EQU     044H    ;VID CUR LOC HI BYTE
0045 =          VCUR    EQU     045H    ;VID CURSOR CHAR
0046 =          VCHUC   EQU     046H    ;VID CHAR UNDER CURSOR
                ;
F000                    ORG     0F000H  ;START OF PROGRAM
                ;
F000 C332F0     VENTER: JMP     VDR     ;JMP TO ROUTINE
                ;
                ;DO VIDEO INIT, CLEAR SCREEN AND HOME CURSOR
                ;
F003 11FEEF     VINIT:  LXI     D,CRTCWR;CRTC WRITE PORT LOC TO DE
F006 060D               MVI     B,0DH   ;B IS A COUNTER
F008 2123F0             LXI     H,VFRMT ;TABLE POINTER
F00B 78         VINIT1: MOV     A,B     ;COUNTER TO A
F00C 32FCEF             STA     CRTCSL  ;WRITE TO CRTC REG SEL PORT
```

```
F00F 7E                 MOV     A,M     ;FORMAT DATA TO A
F010 12                 STAX    D       ;WRITE FORMAT DATA TO CRTC REG
F011 23                 INX     H       ;INC TO NEXT FORMAT DATA
F012 05                 DCR     B       ;DEC REG COUNTER
F013 F20BF0             JP      VINIT1  ;CONT UNTIL B IS NEG
F016 7E                 MOV     A,M     ;CONTROL BYTE TO A
F017 32F8EF             STA     VCTRL   ;WRITE TO VID CTRL REG
F01A 3E7F               MVI     A,CURCH ;CURSOR ASCII CHAR TO A
F01C 324500             STA     VCUR    ;STORE IN SAVE AREA
F01F CD70F0             CALL    VCLR    ;CLEAR SCREEN AND HOME CURSOR
F022 C9                 RET
                ;
                        IF      V80CHAR ;REGISTER PARAMETERS FOR 80 CHAR BOARD
F023 00000820   VFRMT:  DB      00H,00H,08H,20H         ;REG D - REG A
F027 08001818           DB      08H,00H,18H,18H         ;REG 9 - REG 6
F02B 081B0156           DB      08H,1BH,01H,56H         ;REG 5 - REG 2
F02F 506F0C             DB      50H,6FH,0CH             ;REG 1 - CNTL
                        ENDIF
                ;
                        IF      V64CHAR ;REGISTER PARAMETERS FOR 64 CHAR BOARD
                VFRMT:  DB      00H,00H,08H,20H         ;REG D - REG A
                        DB      08H,00H,18H,18H         ;REG 9 - REG 6
                        DB      08H,1BH,01H,46H         ;REG 5 - REG 2
                        DB      40H,5FH,0CH             ;REG 1 - CNTL
                        ENDIF
                ;
                ;
                ;VID DRIVER ROUTINE, CHAR IN C, RETURNS CHAR IN C
                ;
                ;
F032 C5         VDR:    PUSH    B       ;SAVE CHAR IN C
F033 CD38F0             CALL    VDR1    ;DO VDR ROUTINE
F036 C1                 POP     B       ;RESTORE CHAR BACK TO C
F037 C9                 RET
F038 215AF0     VDR1:   LXI     H,VCNTBL;LOAD SEARCH POINTER
F03B CDABF1             CALL    SRCH    ;DO CNTL CHAR SEARCH
F03E FEFF               CPI     0FFH    ;SEE IF ANY FOUND
F040 C2B7F1             JNZ     TBJP    ;GO TO JUMP ROUTINE IF FOUND
F043 79                 MOV     A,C     ;ASCII CHAR TO A
F044 FE20               CPI     ' '     ;SEE IF OTHER CNTL CHAR
F046 D8                 RC              ;RETURN IF SO
F047 CD0BF1             CALL    VWR     ;OUTPUT TO SCREEN
F04A 23                 INX     H       ;INC TO NEXT CUR LOC
F04B CD22F1             CALL    VOFFSC  ;SEE IF OFF SCREEN
F04E DA54F0             JC      VDR2    ;CARRY SET IF OFF SCREEN
F051 C312F1             JMP     VSTOR   ;SAVE CURSOR PARA AND WRITE CUR
F054 2130EF     VDR2:   LXI     H,VIDEND-VSIZE+1 ;SET HL TO START OF LAST LINE
F057 C32EF1             JMP     VSCROL  ;GO TO SCROLL ROUTINE
                ;
F05A 0C         VCNTBL: DB      0CH     ;^L CLEAR SCREEN
F05B 70F0               DW      VCLR    ;
F05D 01                 DB      01H     ;^A HOME
F05E 90F0               DW      VHOME   ;
F060 0D                 DB      0DH     ;^M CARRIAGE RET
F061 9DF0               DW      VCR     ;
```

```
F063 0A                 DB      0AH     ;^J LINE FEED
F064 ACF0               DW      VLF     ;
F066 08                 DB      08H     ;^H BACKSPACE
F067 CAF0               DW      VBKSP   ;
F069 1A                 DB      1AH     ;^Z CURSOR UP
F06A D8F0               DW      VUP     ;
F06C 06                 DB      06H     ;^F CURSOR RIGHT
F06D EEF0               DW      VRT     ;
F06F FF                 DB      0FFH    ;END OF TABLE
                ;
                ;
                ;VID CLEAR SCREEN ROUTINE, CLEARS SCREEN AND HOMES CUR
                ;
                ;
F070 CD96F0     VCLR:   CALL    VHOME1  ;HOME CURSOR
F073 2101E8             LXI     H,VBASE+1 ;START OF RAM + 1 TO HL
F076 3E08               MVI     A,08H   ;BLANK CODE TO A
F078 32F8EF             STA     VCTRL   ;BLANK SCREEN
F07B 3E20       VCLR1:  MVI     A,' '   ;ASCII SPACE TO A
F07D 77                 MOV     M,A     ;SPACE TO LOC HL
F07E 23                 INX     H       ;INC TO NEXT SCREEN LOC
F07F CD22F1             CALL    VOFFSC  ;SEE IF OFF SCREEN
F082 D27BF0             JNC     VCLR1   ;CONT IF NOT
F085 3E20               MVI     A,' '   ;ASCII SPACE TO A
F087 324600             STA     VCHUC   ;BLANK CHAR UNDER CUR
F08A 3E0C               MVI     A,0CH   ;UNBLANK CODE TO A
F08C 32F8EF             STA     VCTRL   ;UNBLANK SCREEN
F08F C9                 RET
                ;
                ;
                ;VID HOME CURSOR ROUTINE, HOMES TO UPPER LEFT
                ;
                ;
F090 3A4600     VHOME:  LDA     VCHUC   ;CHAR UNDER CUR TO A
F093 CD0BF1             CALL    VWR     ;WRITE TO SCREEN
F096 2100E8     VHOME1: LXI     H,VBASE ;HOME POS TO HL
F099 CD12F1             CALL    VSTOR   ;SAVE CUR PARA AND WRITE CUR
F09C C9                 RET
                ;
                ;
                ;VID CARRIAGE RETURN ROUTINE, MOVES CUR TO START OF LINE
                ;
                ;
F09D CD70F1     VCR:    CALL    VSOL    ;FIND START OF LINE
F0A0 C8                 RZ              ;RET IF AT START OF LINE (Z SET)
F0A1 EB                 XCHG            ;SAVE START LOC IN DE
F0A2 3A4600             LDA     VCHUC   ;CHAR UNDER CUR TO A
F0A5 CD0BF1             CALL    VWR     ;WRITE OLD CHAR
F0A8 EB                 XCHG            ;START LOC TO HL
F0A9 C312F1             JMP     VSTOR   ;STORE CUR PARA AND WRITE CUR
                ;
                ;
                ;VID LINE FEED
                ;
                ;
```

```
F0AC 2A4300     VLF:    LHLD    VCADL   ;CUR LOC TO HL
F0AF 115000             LXI     D,VSIZE ;LINE LENGTH TO DE
F0B2 19                 DAD     D       ;ADD DE TO HL
F0B3 EB                 XCHG            ;SAVE NEW CUR LOC IN DE
F0B4 3A4600             LDA     VCHUC   ;CHAR UNDER CUR TO A
F0B7 CD0BF1             CALL    VWR     ;WRITE IT TO SCREEN
F0BA EB                 XCHG            ;NEW CUR LOC TO HL
F0BB CD22F1             CALL    VOFFSC  ;SEE IF OFF SCREEN
F0BE DAC4F0             JC      VLF1    ;JUMP IF OFF
F0C1 C312F1             JMP     VSTOR   ;STORE CUR PARA AND WRITE CUR
F0C4 2A4300     VLF1:   LHLD    VCADL   ;CUR ADD AFTER SCROLL TO HL
F0C7 C32EF1             JMP     VSCROL  ;SCROLL SCREEN
                ;
                ;
                ;VID BACKSPACE ROUTINE, WILL NOT MOVE PAST START OF LINE
                ;
                ;
F0CA CD70F1     VBKSP:  CALL    VSOL    ;SEE IF AT START OF LINE
F0CD C8                 RZ              ;RET IF SO (Z SET)
F0CE 3A4600             LDA     VCHUC   ;CHAR UNDER CUR TO A
F0D1 CD0BF1             CALL    VWR     ;WRITE OLD CHAR
F0D4 2B                 DCX     H       ;DEC HL ONE SPACE
F0D5 C312F1             JMP     VSTOR   ;SAVE CUR PARA AND WRITE CUR
                ;
                ;
                ;VID UP ROUTINE, WILL NOT GO PAST TOP OF PAGE
                ;
                ;
F0D8 2A4300     VUP:    LHLD    VCADL   ;CUR LOC TO HL
F0DB 11B0FF             LXI     D,-VSIZE;NEG LINE LENGTH TO DE
F0DE 19                 DAD     D       ;ADD DE TO HL
F0DF 3EE8               MVI     A,VBASE SHR 8 ;VID RAM STARTING PAGE TO A
F0E1 BC                 CMP     H       ;A - H
F0E2 D0                 RNC             ;OFF OF TOP IF C SET, RET
F0E3 EB                 XCHG            ;STORE NEW CUR LOC IN DE
F0E4 3A4600             LDA     VCHUC   ;CHAR UNDER CUR TO A
F0E7 CD0BF1             CALL    VWR     ;WRITE OLD CHAR
F0EA EB                 XCHG            ;NEW CUR LOC TO HL
F0EB C312F1             JMP     VSTOR   ;SAVE CUR PARA AND WRITE CUR
                ;
                ;
                ;VID CURSOR RIGHT ROUTINE
                ;
                ;
F0EE 3A4600     VRT:    LDA     VCHUC   ;CHAR UNDER CUR TO A
F0F1 CD0BF1             CALL    VWR     ;WRITE OLD CHAR
F0F4 23                 INX     H       ;INC TO NEXT CUR LOC
F0F5 224300             SHLD    VCADL   ;SAVE NEW CUR LOC
F0F8 CD70F1             CALL    VSOL    ;SEE IF IT WRAPPED AROUND
F0FB CA04F1             JZ      VRT1    ;JUMP IF IT DID
F0FE 2A4300             LHLD    VCADL   ;NEW CUR ADD TO HL
F101 C312F1             JMP     VSTOR   ;SAVE CUR PARA AND WRITE CUR
F104 2A4300     VRT1:   LHLD    VCADL   ;NEW CUR ADD TO HL
F107 2B                 DCX     H       ;DEC BACK TO ORIGINAL
F108 C312F1             JMP     VSTOR   ;STORE ORIGINAL PARA AND WRITE CUR
```

```
                        ;
                        ;
                        ;VID WRITE ROUTINE, WRITES A TO CURRENT CUR LOCATION
                        ;
                        ;
F10B 2A4300     VWR:    LHLD    VCADL   ;CUR LOC TO HL
F10E CD92F1             CALL    VOUT    ;A TO LOC HL
F111 C9                 RET
                        ;
                        ;
                        ;VID STORE ROUTINE, SAVES NEW CUR LOCATION (ENTERS IN HL)
                        ;AND SAVES CHAR AT NEW CUR LOC, WRITES CUR CHAR TO NEW
                        ;LOCATION
                        ;
                        ;
F112 224300     VSTOR:  SHLD    VCADL   ;SAVE NEW CUR LOC
F115 CD8DF1             CALL    VIN     ;NEW CHAR UNDER CUR TO A
F118 324600             STA     VCHUC   ;STORE CHAR UNDER CUR
F11B 3A4500             LDA     VCUR    ;CUR CHAR TO A
F11E CD92F1             CALL    VOUT    ;WRITE CURSOR
F121 C9                 RET
                        ;
                        ;
                        ;VID ROUTINE TO SEE IF HL IS POINTING PAST THE END OF
                        ;THE DISPLAYED VID RAM, RETURNS WITH CARRY SET IF SO
                        ;
                        ;
F122 3EEF       VOFFSC: MVI     A,VIDEND SHR 8 ;HI BYTE OF VID RAM END TO A
F124 BC                 CMP     H       ;SEE IF HI BYTE IS ABOVE
F125 D8                 RC              ;RET IF SO
F126 CA2AF1             JZ      VOFFSC1 ;IF =, TEST LO BYTE
F129 C9                 RET
F12A 3E7F       VOFFSC1:MVI     A,VIDEND AND OFFH ;LO BYTE OF RAM END TO A
F12C BD                 CMP     L
F12D C9                 RET             ;CARRY SET IF ABOVE
                        ;
                        ;
                        ;VID SCROLL ROUTINE, SCROLLS SCREEN UP ONE LINE
                        ;
                        ;
F12E E5         VSCROL: PUSH    H       ;SAVE HL
F12F 1100E8             LXI     D,VBASE ;DE POINTS TO START OF VID RAM
F132 2150E8             LXI     H,VBASE+VSIZE ;HL POINTS TO NEXT LINE
F135 013007             LXI     B,23*VSIZE ;NUMBER OF BYTES TO MOVE
F138 CD98F1     VSCWT1: CALL    VSTAT   ;WAIT FOR VERT RETRACE
F13B D238F1             JNC     VSCWT1  ;LOOP
F13E 3E08               MVI     A,08H   ;BLANK CODE TO A
F140 32F8EF             STA     VCTRL   ;BLANK SCREEN
F143 7E         VSCROLO:MOV     A,M     ;THIS CODE DOES THE ACTUAL SCROL
F144 12                 STAX    D       ;DATA UP ONE LINE
F145 13                 INX     D       ;INCRE POINTERS
F146 23                 INX     H
F147 0B                 DCX     B       ;DECRE BYTE COUNTER
F148 AF                 XRA     A       ;ZERO A
F149 B8                 CMP     B       ;SEE IF BC=0
```

```
F14A C243F1             JNZ     VSCROLO ;LOOP IF NOT O
F14D B9                 CMP     C       ;
F14E C243F1             JNZ     VSCROLO ;LOOP IF NOT O
F151 2130EF             LXI     H,VIDEND-VSIZE+1;HL POINTS TO START OF LAST LINE
F154 CD22F1     VSCROL1:CALL    VOFFSC  ;SEE IF OFF SCREEN
F157 DA61F1             JC      VSCROL2 ;JUMP IF OFF
F15A 3E20               MVI     A,' '   ;ASCII BLANK TO A
F15C 77                 MOV     M,A     ;BLANK LAST LINE
F15D 23                 INX     H       ;INC TO NEXT LOC
F15E C354F1             JMP     VSCROL1 ;CONT
F161 CD98F1     VSCROL2:CALL    VSTAT   ;WAIT FOR VERT RETRACE
F164 D261F1             JNC     VSCROL2 ;LOOP
F167 3E0C               MVI     A,0CH   ;UNBLANK CODE TO A
F169 32F8EF             STA     VCTRL   ;UNBLANK SCREEN
F16C E1                 POP     H       ;RESTORE HL (NEW CUR LOC)
F16D C312F1             JMP     VSTOR   ;SAVE CUR PARA AND WRITE CURSOR
                ;
                ;
                ;VID START OF LINE ROUTINE, RETURNS ADDRESS OF THE START OF
                ;THE LINE THE CURSOR IS IN (RETURNED IN HL), Z IS SET
                ;IF THE CURSOR IS AT THE START POINT
                ;
                ;
F170 2100E8     VSOL:   LXI     H,VBASE ;BASE ADD TO HL
F173 015000             LXI     B,VSIZE ;LINE SIZE TO BC
F176 54         VSOL1:  MOV     D,H     ;STORE HL IN DE
F177 5D                 MOV     E,L
F178 09                 DAD     B       ;INC HL TO START OF NEXT LINE
F179 3A4400             LDA     VCADH   ;CUR LOC HI BYTE TO A
F17C BC                 CMP     H       ;HI BYTE - H
F17D DA8BF1             JC      VSOL2   ;HL > CUR LOC, USE LAST SOL IN DE
F180 C276F1             JNZ     VSOL1   ;TRY NEXT LINE
F183 3A4300             LDA     VCADL   ;CHECK LO BYTE
F186 BD                 CMP     L       ;LO BYTE - L
F187 C8                 RZ              ;CUR AT START OF LINE, RET
F188 D276F1             JNC     VSOL1   ;TRY NEXT LINE
F18B EB         VSOL2:  XCHG            ;LAST SOL TO HL FROM DE
F18C C9                 RET
                ;
                ;
                ;VID RAM READ ROUTINE, READS RAM POINTED TO BY HL
                ;DURING SCREEN RETRACE, RETURNS CHAR IN A
                ;
                ;
F18D CD98F1     VIN:    CALL    VSTAT   ;WAIT FOR RETRACE
F190 7E                 MOV     A,M     ;READ IF B6=0=RETRACE
F191 C9                 RET
                ;
                ;
                ;VID RAM WRITE ROUTINE, WRITES A TO RAM POINTED TO
                ;BY HL DURING SCREEN RETRACE
                ;
                ;
F192 4F         VOUT:   MOV     C,A     ;SAVE ASCII IN C
F193 CD98F1             CALL    VSTAT   ;WAIT FOR RETRACE
```

```
F196 71                 MOV     M,C     ;DATA OUT
F197 C9                 RET
                ;
                ;
                ;VID STATUS ROUTINE, WAITS FOR START OF A HORZ RETRACE THEN
                ;RETURNS, OR RETURNS IMMED IF IN A VERT RETRACE WITH CARRY SET
                ;B6=1(DISPLAY) =0(RETRACE), B7=1(1MS AT START OF VERT RETRACE)
                ;
                ;
F198 3AF8EF     VSTAT:  LDA     VCTRL   ;READ STATUS
F19B 17                 RAL             ;SHIFT VERT STAT TO CARRY
F19C D8                 RC              ;RETURN IF IN A VERT RETRACE
F19D 17                 RAL             ;SHIFT B6 TO CARRY
F19E D298F1             JNC     VSTAT   ;WAIT UNTIL NEXT RETRACE
F1A1 3AF8EF     VSTAT1: LDA     VCTRL   ;READ STATUS AGAIN
F1A4 17                 RAL             ;SHIFT VERT STATUS TO CARRY
F1A5 D8                 RC              ;RETURN IF IN A VERT RETRACE
F1A6 17                 RAL             ;SHIFT B6 TO CARRY
F1A7 DAA1F1             JC      VSTAT1  ;WAIT FOR RETRACE
F1AA C9                 RET
                ;
                ;
                ;SEARCH ROUTINE, HL POINTS TO TABLE, INPUT IS IN C
                ;
                ;
F1AB 7E         SRCH:   MOV     A,M     ;GET DATA FROM TABLE
F1AC FEFF               CPI     OFFH    ;SEE IF END OF TABLE
F1AE C8                 RZ              ;RETURN IF SO
F1AF B9                 CMP     C       ;COMPARE WITH INPUT
F1B0 C8                 RZ              ;RETURN IF =
F1B1 23                 INX     H
F1B2 23                 INX     H
F1B3 23                 INX     H
F1B4 C3ABF1             JMP     SRCH    ;CHECK NEXT DATA
F1B7 23         TBJP:   INX     H       ;INCRE TO L JUMP BYTE
F1B8 5E                 MOV     E,M     ;PUT IN E
F1B9 23                 INX     H       ;INCRE TO H JUMP BYTE
F1BA 56                 MOV     D,M     ;PUT IN D
F1BB EB                 XCHG            ;PUT DE INTO HL
F1BC E9                 PCHL            ;JUMP TO HL
                ;
F1BD                    END
```

```
                ;***********************************************
                ;***                                         ***
                ;***  8085/Z80 Video Driver Routine          ***
                ;***  for the STD VID64/80 board             ***
                ;***                                         ***
                ;***  64 character version 1.1               ***
                ;***  COPYRIGHT 1982 VersaLogic              ***
                ;***                                         ***
                ;***********************************************
                ;
                ;VDR ENTRY POINT=F000H, ASCII CHARACTER IN C REG
                ;VDR INIT ENTRY POINT=F003H
                ;
                ;STD VID64/80 BOARD ADDRESSED AT LOC E800H
                ;RAM WORK AREA AT 0043H-0046H, 4 BYTES
                ;
                ;
FFFF =          YES     EQU     OFFFFH  ;SET VALUES FOR IF STATEMENTS
0000 =          NO      EQU     00000H  ;
                ;
0000 =          V80CHAR EQU     NO      ;80 CHARACTER BOARD?
FFFF =          V64CHAR EQU     YES     ;64 CHARACTER BOARD?
                ;
                        IF      V80CHAR
                VSIZE   EQU     80      ;SET VSIZE FOR 80 CHAR BOARD
                        ENDIF
                ;
                        IF      V64CHAR
0040 =          VSIZE   EQU     64      ;SET VSIZE FOR 64 CHAR BOARD
                        ENDIF
                ;
E800 =          VBASE   EQU     0E800H      ;START OF STD VID 64/80 BOARD
EDFF =          VIDEND  EQU     VBASE+(24*VSIZE)-1 ;END OF DISPLAYED VID RAM
EFF8 =          VCTRL   EQU     VBASE+7F8H ;VID CONTROL AND STATUS ADD
EFFC =          CRTCSL  EQU     VBASE+7FCH ;CRTC REG SELECT ADD
EFFE =          CRTCWR  EQU     VBASE+7FEH ;CRTC WRITE PORT ADD
                ;
007F =          CURCH   EQU     7FH     ;CURSOR CHARACTER
                ;
0043 =          VCADL   EQU     043H    ;VID CUR LOC LO BYTE
0044 =          VCADH   EQU     044H    ;VID CUR LOC HI BYTE
0045 =          VCUR    EQU     045H    ;VID CURSOR CHAR
0046 =          VCHUC   EQU     046H    ;VID CHAR UNDER CURSOR
                ;
F000                    ORG     0F000H  ;START OF PROGRAM
                ;
F000 C332F0     VENTER: JMP     VDR     ;JMP TO ROUTINE
                ;
                ;DO VIDEO INIT, CLEAR SCREEN AND HOME CURSOR
                ;
F003 11FEEF     VINIT:  LXI     D,CRTCWR;CRTC WRITE PORT LOC TO DE
F006 060D               MVI     B,0DH   ;B IS A COUNTER
F008 2123F0             LXI     H,VFRMT ;TABLE POINTER
F00B 78         VINIT1: MOV     A,B     ;COUNTER TO A
F00C 32FCEF             STA     CRTCSL  ;WRITE TO CRTC REG SEL PORT
```

```
F00F 7E                 MOV     A,M     ;FORMAT DATA TO A
F010 12                 STAX    D       ;WRITE FORMAT DATA TO CRTC REG
F011 23                 INX     H       ;INC TO NEXT FORMAT DATA
F012 05                 DCR     B       ;DEC REG COUNTER
F013 F20BF0             JP      VINIT1  ;CONT UNTIL B IS NEG
F016 7E                 MOV     A,M     ;CONTROL BYTE TO A
F017 32F8EF             STA     VCTRL   ;WRITE TO VID CTRL REG
F01A 3E7F               MVI     A,CURCH ;CURSOR ASCII CHAR TO A
F01C 324500             STA     VCUR    ;STORE IN SAVE AREA
F01F CD70F0             CALL    VCLR    ;CLEAR SCREEN AND HOME CURSOR
F022 C9                 RET
                ;
                        IF      V80CHAR ;REGISTER PARAMETERS FOR 80 CHAR BOARD
                VFRMT:  DB      00H,00H,08H,20H         ;REG D - REG A
                        DB      08H,00H,18H,18H         ;REG 9 - REG 6
                        DB      08H,1BH,01H,56H         ;REG 5 - REG 2
                        DB      50H,6FH,0CH             ;REG 1 - CNTL
                        ENDIF
                ;
                        IF      V64CHAR ;REGISTER PARAMETERS FOR 64 CHAR BOARD
F023 00000820   VFRMT:  DB      00H,00H,08H,20H         ;REG D - REG A
F027 08001818           DB      08H,00H,18H,18H         ;REG 9 - REG 6
F02B 081B0146           DB      08H,1BH,01H,46H         ;REG 5 - REG 2
F02F 405F0C             DB      40H,5FH,0CH             ;REG 1 - CNTL
                        ENDIF
                ;
                ;
                ;VID DRIVER ROUTINE, CHAR IN C, RETURNS CHAR IN C
                ;
                ;
F032 C5         VDR:    PUSH    B       ;SAVE CHAR IN C
F033 CD38F0             CALL    VDR1    ;DO VDR ROUTINE
F036 C1                 POP     B       ;RESTORE CHAR BACK TO C
F037 C9                 RET
F038 215AF0     VDR1:   LXI     H,VCNTBL;LOAD SEARCH POINTER
F03B CDABF1             CALL    SRCH    ;DO CNTL CHAR SEARCH
F03E FEFF               CPI     0FFH    ;SEE IF ANY FOUND
F040 C2B7F1             JNZ     TBJP    ;GO TO JUMP ROUTINE IF FOUND
F043 79                 MOV     A,C     ;ASCII CHAR TO A
F044 FE20               CPI     ' '     ;SEE IF OTHER CNTL CHAR
F046 D8                 RC              ;RETURN IF SO
F047 CD0BF1             CALL    VWR     ;OUTPUT TO SCREEN
F04A 23                 INX     H       ;INC TO NEXT CUR LOC
F04B CD22F1             CALL    VOFFSC  ;SEE IF OFF SCREEN
F04E DA54F0             JC      VDR2    ;CARRY SET IF OFF SCREEN
F051 C312F1             JMP     VSTOR   ;SAVE CURSOR PARA AND WRITE CUR
F054 210ED0     VDR2:   LXI     H,VIDEND-VSIZE+1 ;SET HL TO START OF LAST LINE
F057 C32EF1             JMP     VSCROL  ;GO TO SCROLL ROUTINE
                ;
F05A 0C         VCNTBL: DB      0CH     ;^L CLEAR SCREEN
F05B 70F0               DW      VCLR    ;
F05D 01                 DB      01H     ;^A HOME
F05E 90F0               DW      VHOME   ;
F060 0D                 DB      0DH     ;^M CARRIAGE RET
F061 9DF0               DW      VCR     ;
```

```
F063 0A                 DB      0AH     ;^J LINE FEED
F064 ACF0               DW      VLF     ;
F066 08                 DB      08H     ;^H BACKSPACE
F067 CAF0               DW      VBKSP   ;
F069 1A                 DB      1AH     ;^Z CURSOR UP
F06A D8F0               DW      VUP     ;
F06C 06                 DB      06H     ;^F CURSOR RIGHT
F06D EEF0               DW      VRT     ;
F06F FF                 DB      OFFH    ;END OF TABLE
                ;
                ;
                ;VID CLEAR SCREEN ROUTINE, CLEARS SCREEN AND HOMES CUR
                ;
                ;
F070 CD96F0     VCLR:   CALL    VHOME1  ;HOME CURSOR
F073 2101E8             LXI     H,VBASE+1 ;START OF RAM + 1 TO HL
F076 3E08               MVI     A,08H   ;BLANK CODE TO A
F078 32F8EF             STA     VCTRL   ;BLANK SCREEN
F07B 3E20       VCLR1:  MVI     A,' '   ;ASCII SPACE TO A
F07D 77                 MOV     M,A     ;SPACE TO LOC HL
F07E 23                 INX     H       ;INC TO NEXT SCREEN LOC
F07F CD22F1             CALL    VOFFSC  ;SEE IF OFF SCREEN
F082 D27BF0             JNC     VCLR1   ;CONT IF NOT
F085 3E20               MVI     A,' '   ;ASCII SPACE TO A
F087 324600             STA     VCHUC   ;BLANK CHAR UNDER CUR
F08A 3E0C               MVI     A,0CH   ;UNBLANK CODE TO A
F08C 32F8EF             STA     VCTRL   ;UNBLANK SCREEN
F08F C9                 RET
                ;
                ;
                ;VID HOME CURSOR ROUTINE, HOMES TO UPPER LEFT
                ;
                ;
F090 3A4600     VHOME:  LDA     VCHUC   ;CHAR UNDER CUR TO A
F093 CD0BF1             CALL    VWR     ;WRITE TO SCREEN
F096 2100E8     VHOME1: LXI     H,VBASE ;HOME POS TO HL
F099 CD12F1             CALL    VSTOR   ;SAVE CUR PARA AND WRITE CUR
F09C C9                 RET
                ;
                ;
                ;VID CARRIAGE RETURN ROUTINE, MOVES CUR TO START OF LINE
                ;
                ;
F09D CD70F1     VCR:    CALL    VSOL    ;FIND START OF LINE
F0A0 C8                 RZ              ;RET IF AT START OF LINE (Z SET)
F0A1 EB                 XCHG            ;SAVE START LOC IN DE
F0A2 3A4600             LDA     VCHUC   ;CHAR UNDER CUR TO A
F0A5 CD0BF1             CALL    VWR     ;WRITE OLD CHAR
F0A8 EB                 XCHG            ;START LOC TO HL
F0A9 C312F1             JMP     VSTOR   ;STORE CUR PARA AND WRITE CUR
                ;
                ;
                ;VID LINE FEED
                ;
                ;
```

```
F0AC 2A4300    VLF:    LHLD    VCADL   ;CUR LOC TO HL
F0AF 114000            LXI     D,VSIZE ;LINE LENGTH TO DE
F0B2 19                DAD     D       ;ADD DE TO HL
F0B3 EB                XCHG            ;SAVE NEW CUR LOC IN DE
F0B4 3A4600            LDA     VCHUC   ;CHAR UNDER CUR TO A
F0B7 CD0BF1            CALL    VWR     ;WRITE IT TO SCREEN
F0BA EB                XCHG            ;NEW CUR LOC TO HL
F0BB CD22F1            CALL    VOFFSC  ;SEE IF OFF SCREEN
F0BE DAC4F0            JC      VLF1    ;JUMP IF OFF
F0C1 C312F1            JMP     VSTOR   ;STORE CUR PARA AND WRITE CUR
F0C4 2A4300    VLF1:   LHLD    VCADL   ;CUR ADD AFTER SCROLL TO HL
F0C7 C32EF1            JMP     VSCROL  ;SCROLL SCREEN
                ;
                ;
                ;VID BACKSPACE ROUTINE, WILL NOT MOVE PAST START OF LINE
                ;
                ;
F0CA CD70F1    VBKSP:  CALL    VSOL    ;SEE IF AT START OF LINE
F0CD C8                RZ              ;RET IF SO (Z SET)
F0CE 3A4600            LDA     VCHUC   ;CHAR UNDER CUR TO A
F0D1 CD0BF1            CALL    VWR     ;WRITE OLD CHAR
F0D4 2B                DCX     H       ;DEC HL ONE SPACE
F0D5 C312F1            JMP     VSTOR   ;SAVE CUR PARA AND WRITE CUR
                ;
                ;
                ;VID UP ROUTINE, WILL NOT GO PAST TOP OF PAGE
                ;
                ;
F0D8 2A4300    VUP:    LHLD    VCADL   ;CUR LOC TO HL
F0DB 11C0FF            LXI     D,-VSIZE;NEG LINE LENGTH TO DE
F0DE 19                DAD     D       ;ADD DE TO HL
F0DF 3EE8              MVI     A,VBASE SHR 8 ;VID RAM STARTING PAGE TO A
F0E1 BC                CMP     H       ;A - H
F0E2 D0                RNC             ;OFF OF TOP IF C SET, RET
F0E3 EB                XCHG            ;STORE NEW CUR LOC IN DE
F0E4 3A4600            LDA     VCHUC   ;CHAR UNDER CUR TO A
F0E7 CD0BF1            CALL    VWR     ;WRITE OLD CHAR
F0EA EB                XCHG            ;NEW CUR LOC TO HL
F0EB C312F1            JMP     VSTOR   ;SAVE CUR PARA AND WRITE CUR
                ;
                ;
                ;VID CURSOR RIGHT ROUTINE
                ;
                ;
F0EE 3A4600    VRT:    LDA     VCHUC   ;CHAR UNDER CUR TO A
F0F1 CD0BF1            CALL    VWR     ;WRITE OLD CHAR
F0F4 23                INX     H       ;INC TO NEXT CUR LOC
F0F5 224300            SHLD    VCADL   ;SAVE NEW CUR LOC
F0F8 CD70F1            CALL    VSOL    ;SEE IF IT WRAPPED AROUND
F0FB CA04F1            JZ      VRT1    ;JUMP IF IT DID
F0FE 2A4300            LHLD    VCADL   ;NEW CUR ADD TO HL
F101 C312F1            JMP     VSTOR   ;SAVE CUR PARA AND WRITE CUR
F104 2A4300    VRT1:   LHLD    VCADL   ;NEW CUR ADD TO HL
F107 2B                DCX     H       ;DEC BACK TO ORIGINAL
F108 C312F1            JMP     VSTOR   ;STORE ORIGINAL PARA AND WRITE CUR
```

```
                        ;
                        ;
                        ;VID WRITE ROUTINE, WRITES A TO CURRENT CUR LOCATION
                        ;
                        ;
F10B 2A4300     VWR:    LHLD    VCADL   ;CUR LOC TO HL
F10E CD92F1             CALL    VOUT    ;A TO LOC HL
F111 C9                 RET
                        ;
                        ;
                        ;VID STORE ROUTINE, SAVES NEW CUR LOCATION (ENTERS IN HL)
                        ;AND SAVES CHAR AT NEW CUR LOC, WRITES CUR CHAR TO NEW
                        ;LOCATION
                        ;
                        ;
F112 224300     VSTOR:  SHLD    VCADL   ;SAVE NEW CUR LOC
F115 CD8DF1             CALL    VIN     ;NEW CHAR UNDER CUR TO A
F118 324600             STA     VCHUC   ;STORE CHAR UNDER CUR
F11B 3A4500             LDA     VCUR    ;CUR CHAR TO A
F11E CD92F1             CALL    VOUT    ;WRITE CURSOR
F121 C9                 RET
                        ;
                        ;
                        ;VID ROUTINE TO SEE IF HL IS POINTING PAST THE END OF
                        ;THE DISPLAYED VID RAM, RETURNS WITH CARRY SET IF SO
                        ;
                        ;
F122 3EED       VOFFSC: MVI     A,VIDEND SHR 8 ;HI BYTE OF VID RAM END TO A
F124 BC                 CMP     H       ;SEE IF HI BYTE IS ABOVE
F125 D8                 RC              ;RET IF SO
F126 CA2AF1             JZ      VOFFSC1 ;IF =, TEST LO BYTE
F129 C9                 RET
F12A 3EFF       VOFFSC1:MVI     A,VIDEND AND OFFH ;LO BYTE OF RAM END TO A
F12C BD                 CMP     L
F12D C9                 RET             ;CARRY SET IF ABOVE
                        ;
                        ;
                        ;VID SCROLL ROUTINE, SCROLLS SCREEN UP ONE LINE
                        ;
                        ;
F12E E5         VSCROL: PUSH    H       ;SAVE HL
F12F 1100E8             LXI     D,VBASE ;DE POINTS TO START OF VID RAM
F132 2140E8             LXI     H,VBASE+VSIZE ;HL POINTS TO NEXT LINE
F135 01C005             LXI     B,23*VSIZE ;NUMBER OF BYTES TO MOVE
F138 CD98F1     VSCWT1: CALL    VSTAT   ;WAIT FOR VERT RETRACE
F13B D238F1             JNC     VSCWT1  ;LOOP
F13E 3E08               MVI     A,08H   ;BLANK CODE TO A
F140 32F8EF             STA     VCTRL   ;BLANK SCREEN
F143 7E         VSCROLO:MOV     A,M     ;THIS CODE DOES THE ACTUAL SCROL
F144 12                 STAX    D       ;DATA UP ONE LINE
F145 13                 INX     D       ;INCRE POINTERS
F146 23                 INX     H
F147 0B                 DCX     B       ;DECRE BYTE COUNTER
F148 AF                 XRA     A       ;ZERO A
F149 B8                 CMP     B       ;SEE IF BC=0
```

```
F14A C243F1             JNZ     VSCROLO ;LOOP IF NOT O
F14D B9                 CMP     C       ;
F14E C243F1             JNZ     VSCROLO ;LOOP IF NOT O
F151 21COED             LXI     H,VIDEND-VSIZE+1;HL POINTS TO START OF LAST LINE
F154 CD22F1     VSCROL1:CALL    VOFFSC  ;SEE IF OFF SCREEN
F157 DA61F1             JC      VSCROL2 ;JUMP IF OFF
F15A 3E20               MVI     A,' '   ;ASCII BLANK TO A
F15C 77                 MOV     M,A     ;BLANK LAST LINE
F15D 23                 INX     H       ;INC TO NEXT LOC
F15E C354F1             JMP     VSCROL1 ;CONT
F161 CD98F1     VSCROL2:CALL    VSTAT   ;WAIT FOR VERT RETRACE
F164 D261F1             JNC     VSCROL2 ;LOOP
F167 3EOC               MVI     A,OCH   ;UNBLANK CODE TO A
F169 32F8EF             STA     VCTRL   ;UNBLANK SCREEN
F16C E1                 POP     H       ;RESTORE HL (NEW CUR LOC)
F16D C312F1             JMP     VSTOR   ;SAVE CUR PARA AND WRITE CURSOR
                ;
                ;
                ;VID START OF LINE ROUTINE, RETURNS ADDRESS OF THE START OF
                ;THE LINE THE CURSOR IS IN (RETURNED IN HL), Z IS SET
                ;IF THE CURSOR IS AT THE START POINT
                ;
                ;
F170 2100E8     VSOL:   LXI     H,VBASE ;BASE ADD TO HL
F173 014000             LXI     B,VSIZE ;LINE SIZE TO BC
F176 54         VSOL1:  MOV     D,H     ;STORE HL IN DE
F177 5D                 MOV     E,L
F178 09                 DAD     B       ;INC HL TO START OF NEXT LINE
F179 3A4400             LDA     VCADH   ;CUR LOC HI BYTE TO A
F17C BC                 CMP     H       ;HI BYTE - H
F17D DA8BF1             JC      VSOL2   ;HL > CUR LOC, USE LAST SOL IN DE
F180 C276F1             JNZ     VSOL1   ;TRY NEXT LINE
F183 3A4300             LDA     VCADL   ;CHECK LO BYTE
F186 BD                 CMP     L       ;LO BYTE - L
F187 C8                 RZ              ;CUR AT START OF LINE, RET
F188 D276F1             JNC     VSOL1   ;TRY NEXT LINE
F18B EB         VSOL2:  XCHG            ;LAST SOL TO HL FROM DE
F18C C9                 RET
                ;
                ;
                ;VID RAM READ ROUTINE, READS RAM POINTED TO BY HL
                ;DURING SCREEN RETRACE, RETURNS CHAR IN A
                ;
                ;
F18D CD98F1     VIN:    CALL    VSTAT   ;WAIT FOR RETRACE
F190 7E                 MOV     A,M     ;READ IF B6=O=RETRACE
F191 C9                 RET
                ;
                ;
                ;VID RAM WRITE ROUTINE, WRITES A TO RAM POINTED TO
                ;BY HL DURING SCREEN RETRACE
                ;
                ;
F192 4F         VOUT:   MOV     C,A     ;SAVE ASCII IN C
F193 CD98F1             CALL    VSTAT   ;WAIT FOR RETRACE
```

```
F196 71                 MOV     M,C     ;DATA OUT
F197 C9                 RET
                ;
                ;
                ;VID STATUS ROUTINE, WAITS FOR START OF A HORZ RETRACE THEN
                ;RETURNS, OR RETURNS IMMED IF IN A VERT RETRACE WITH CARRY SET
                ;B6=1(DISPLAY) =0(RETRACE), B7=1(1MS AT START OF VERT RETRACE)
                ;
                ;
F198 3AF8EF     VSTAT:  LDA     VCTRL   ;READ STATUS
F19B 17                 RAL             ;SHIFT VERT STAT TO CARRY
F19C D8                 RC              ;RETURN IF IN A VERT RETRACE
F19D 17                 RAL             ;SHIFT B6 TO CARRY
F19E D298F1             JNC     VSTAT   ;WAIT UNTIL NEXT RETRACE
F1A1 3AF8EF     VSTAT1: LDA     VCTRL   ;READ STATUS AGAIN
F1A4 17                 RAL             ;SHIFT VERT STATUS TO CARRY
F1A5 D8                 RC              ;RETURN IF IN A VERT RETRACE
F1A6 17                 RAL             ;SHIFT B6 TO CARRY
F1A7 DAA1F1             JC      VSTAT1  ;WAIT FOR RETRACE
F1AA C9                 RET
                ;
                ;
                ;SEARCH ROUTINE, HL POINTS TO TABLE, INPUT IS IN C
                ;
                ;
F1AB 7E         SRCH:   MOV     A,M     ;GET DATA FROM TABLE
F1AC FEFF               CPI     OFFH    ;SEE IF END OF TABLE
F1AE C8                 RZ              ;RETURN IF SO
F1AF B9                 CMP     C       ;COMPARE WITH INPUT
F1B0 C8                 RZ              ;RETURN IF =
F1B1 23                 INX     H
F1B2 23                 INX     H
F1B3 23                 INX     H
F1B4 C3ABF1             JMP     SRCH    ;CHECK NEXT DATA
F1B7 23         TBJP:   INX     H       ;INCRE TO L JUMP BYTE
F1B8 5E                 MOV     E,M     ;PUT IN E
F1B9 23                 INX     H       ;INCRE TO H JUMP BYTE
F1BA 56                 MOV     D,M     ;PUT IN D
F1BB EB                 XCHG            ;PUT DE INTO HL
F1BC E9                 PCHL            ;JUMP TO HL
                ;
F1BD                    END
```