

MOSTEK®

USING MK3807 VCU IN A MICROPROCESSOR ENVIRONMENT

Application Note

INTRODUCTION

MK3807, the programmable CRT Video Control Unit (VCU), is a user programmable 40-pin n-channel MOS/LSI chip containing the logic functions required to generate all the timing signals for the formatting and presentation of interlaced or non-interlaced video data on a standard or non-standard CRT monitor.

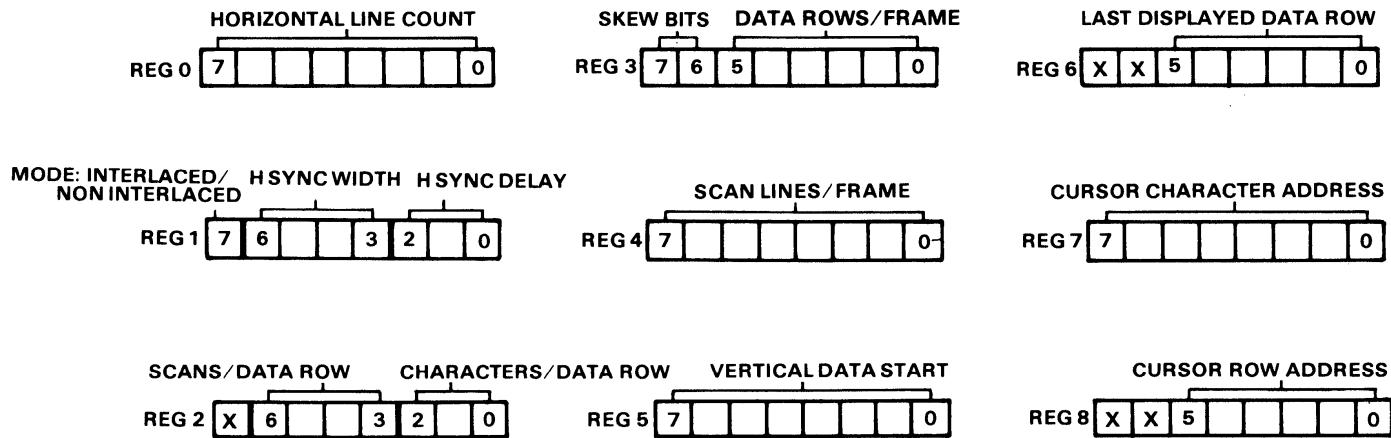
All the formatting, such as horizontal, vertical, and composite sync, characters per data row and per frame are totally user programmable. The data row counter has been designed to facilitate scrolling.

Programming is accomplished by loading seven 8 bit control registers directly off an 8 bit bidirectional data bus. Four register address lines and a chip enable line provide complete microprocessor compatibility for program controlled set up. The device can also be "self loaded" via an external PROM tied on the data bus. (See Figure 1).

In addition to the seven control registers, two additional registers are provided to store the cursor character and row addresses for generation of the cursor video signal. The contents of these two registers can be read out onto the bus for update by the program or used by the microprocessor as two memory locations. (See Figure 2).

BIT ASSIGNMENT

Figure 2



REGISTER 0

This 8 bit register contains the number of character times for 1 horizontal period of the TV raster scan. For example, using American Standard Television ($63.5 \mu s$ per line) at a character time of 500 ns, the value for this register would be $63.5 \text{ divided by } .5 = 127$. The number in this register is normally 1.25 times the number of characters per line displayed on this screen. The value loaded into this register is the binary equivalent of 126 (127-1). Since character times are counted from zero instead of one, the value loaded into this register is one less than the actual number of character times. (Refer to Figure 3 for timing diagrams).

REGISTER 1

This register contains 3 fields of information. The most significant bit (7) is the interlace bit. If this bit is set to a 1, Interlace mode is indicated; if set to a 0, Non-Interlace mode is indicated. The next 4 bits (6-3) define the number of character times for the width of the horizontal sync pulse. For example, using American Standard Television ($4.5 \mu s$) and a character time of 500 ns indicates that it would require 9 character times, therefore the binary equivalent 9 would be loaded in these bits. The least significant 3 bits (2-0) are used to specify the horizontal sync delay. This is commonly called the Front Porch and is the period between the end of active video to the beginning of the horizontal sync pulse. The value here is not critical and can be used to position the video horizontally on the screen.

REGISTER 2

This register contains both the number of characters to be displayed per line as well as the number of scans per character. Bit 7 is not used ($B7 = X$). Bits 6 through 3 define the number of scans per character. For example, using a 7X9 dot matrix character generator, the normal number of scans might be 12. Therefore, using 12 scans per character, the binary equivalent of eleven (12-1) is inserted into this field. The least significant 3 bits (2-0) contain a 3 bit code

which defines the number of characters per line. The VCU is pre-programmed for 20, 32, 40, 64, 72, 80, 96, and 132 characters per line. The 3 bit binary number used in this field determines the particular format, for example, 80 characters being the 6th value would be coded as a binary 5 (101).

CHARACTERS/DATA ROW

DB2	DB1	DB0	
0	0	0	= 20
0	0	1	= 32
0	1	0	= 40
0	1	1	= 64
1	0	0	= 72
1	0	1	= 80
1	1	0	= 96
1	1	1	= 132

REGISTER 3

This register contains both the propagation delay compensation field (skew bits) as well as the data row fields. Bits 7 and 6 are used to adjust the blanking, cursor position and sync delay so as to compensate for either 0, 1 or 2 character time propagation delays of the character generator and the frame buffer RAM.

SKEW BITS

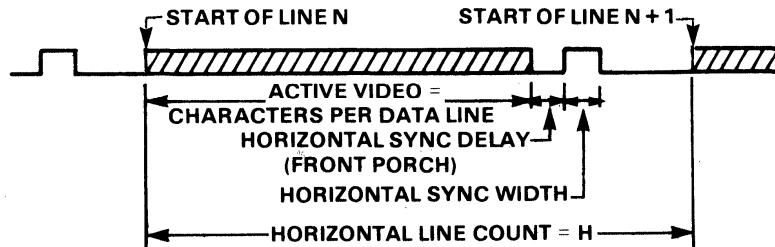
DB7	DB6	Sync/Blank Delay	Cursor Delay
		(Character Times)	
0	0	0	0
1	0	1	0
0	1	2	1
1	1	2	2

The 6 least significant bits (5-0) define the number of data rows to be displayed on the screen. The number of rows begins at 000000 (single row) and continues to 111111 (64 rows).

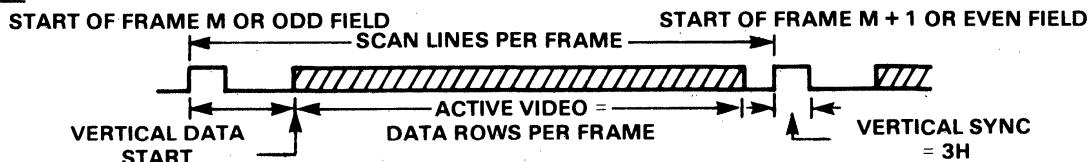
HORIZONTAL AND VERTICLE TIMING

Figure 3

HORIZONTAL TIMING

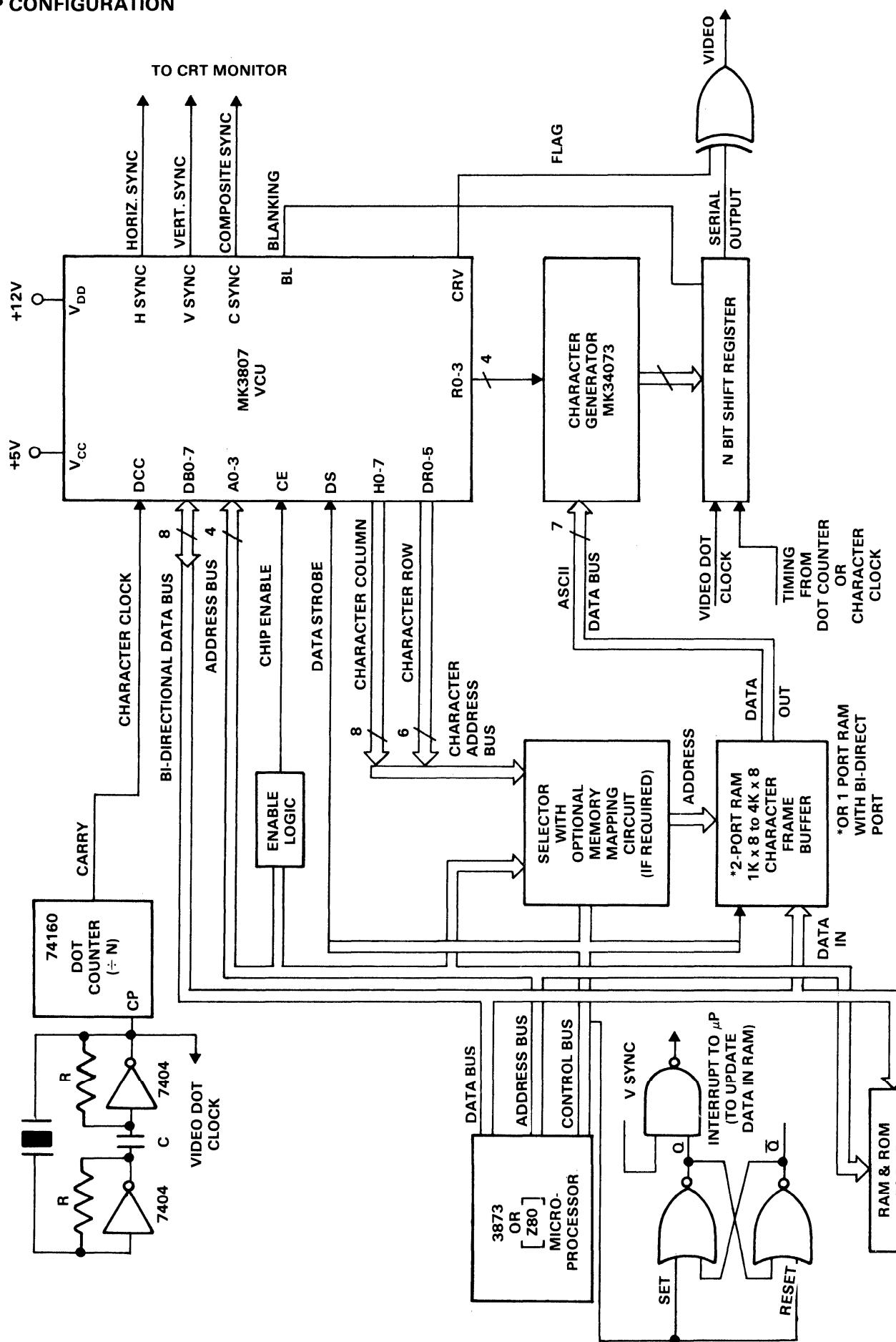


VERTICAL TIMING



VCU μ P CONFIGURATION

Figure 4



VI
Z80
MICRO-
COMPUTER
APPLICATION
NOTES

REGISTER 4

This 8 bit register defines the number of raster lines in the field (frame). Care should be taken when programming this register to make sure that the product of the scans per data row times the number of data rows is less than the number of raster scans. There are 2 methods of programming this register. In the interlaced mode subtract 513 from the number of raster lines desired and divide by 2. For example, for 525 scans, the register should contain the number 6. In the non-interlaced mode subtract the number 256 from the desired number of raster lines and divide by 2. For example, for 262 raster lines, the value is 3.

REGISTER 5

This register defines the number of raster lines between the beginning of the vertical sync pulse and the start of the first data row being displayed. Typically, values of 20 or 21 lines are used. Higher values can be used to position data lower on the screen to a maximum 255. This is called Vertical Data Start and is the sum of Vertical Sync and Vertical Scan Delay.

REGISTER 6

The least significant 6 bits (5-0) of this register define the last data row to be displayed on the screen. Bits 7 and 6 are not used. This feature is useful for both scrolling and positioning of data. For example, if the display was set for 24 data rows, normally row 0 would be on top of the screen and row 23 would be at the bottom. If the scroll register (register 6) contained the number 15, then row 15 would be at the bottom and row 16 would be at the top of the screen. Row 23 and row 0 would be contiguous in the middle of the screen.

REGISTER 7

This 8 bit register contains the character number at which the cursor is to be addressed. For example, if the last character of an 80 character per line display were to be censored, the binary equivalent of 79 would be in this register.

REGISTER 8

The least significant 6 bits (5-0) of this register define the data row for the cursor; similar to Register 7.

BASIC DISPLAY CONFIGURATION

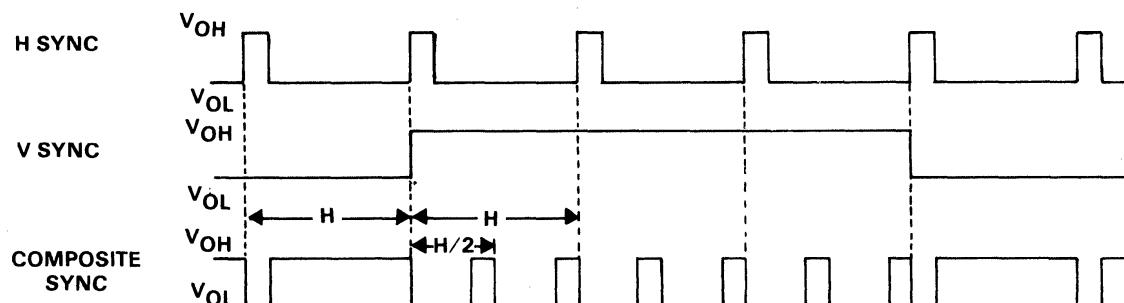
Figure 4 shows the basic configuration for a Bus Oriented, microprocessor based, CRT display system utilizing Mostek's MK3807, the Programmable CRT Video Control Unit (VCU). Either a standard or a non-standard CRT monitor may be used. The user programmable VCU provides Horizontal Sync, Vertical Sync and Composite Sync with serrations, to the monitor's sync deflection circuitry. (Figure 5 shows the composite sync timing). A serial output character generator provides video dot clock frequency data to the Z axis video input of the monitor.

In addition to the VCU, character generator, and shift register, the display system requires a crystal oscillator and a dot counter, typically consisting of two gates of a 7404 and a crystal as well as a 74160 (or equivalent) dot counter. The dot counter divisor (N) is set for the number of horizontal bits in the character plus the number of dots desired for spacing (i.e., for a 7 bit wide character + 2 dots of spacing N = 9). The carry output of the dot counter pulses once per character (character clock) and is fed into the MK3807 DCC (pin 12) input. This enables the VCU to keep track of the character positions as well as generate the entire video timing chain. At the same time the output of the oscillator is fed into the video dot clock input of the shift register of the Video Signal Generator.

An 8 bit bidirectional Data Bus (DB0-DB7), a 4 bit Address Bus (A0-A3), a Chip Enable and a Data Strobe are used in programming the VCU. These buses connect to the microprocessor Data Bus and Address Bus. The VCU appears to the microprocessor as 16 memory or I/O locations. Page logic (high order address bit decoder) connects the Address Bus to the Chip Enable (CE) thereby determining where in the microprocessor memory space the VCU will be located. The Data Strobe (DS) signal is connected to the microprocessor Control Bus. This is used to read or write via the Data Bus, as well as to activate control functions.

COMPOSITE SYNC TIMING DIAGRAM

Figure 5



The VCU raster scan counter outputs (R0-R3) are connected directly to the raster line address inputs of the character generator. This 4 bit address indicates which raster line of the selected character is to be parallel loaded into the shift register. The bit pattern, along with the additional blank spaces, is then shifted out of the video output at the video dot clock rate. The blanking signal can be connected to retrace blanking logic to provide both horizontal and vertical blanking of the video signal to the CRT monitor. The load/shift signals for character generator logic can be derived from the outputs of the dot counter (74160) or taken directly from the character clock (DCC, pin 12 of 3807).

HOW TO USE ROW-COLUMN ADDRESSING

The VCU outputs the character position via the character counter outputs (H0-H7) and the data row counter outputs (DR0-DR5). These outputs define the character column and row location. They are used to address a character frame buffer RAM in which the frame image is stored. Since the VCU keeps counting horizontal addresses (H0-H7) during both horizontal and vertical blanking, dynamic RAMs may be refreshed.

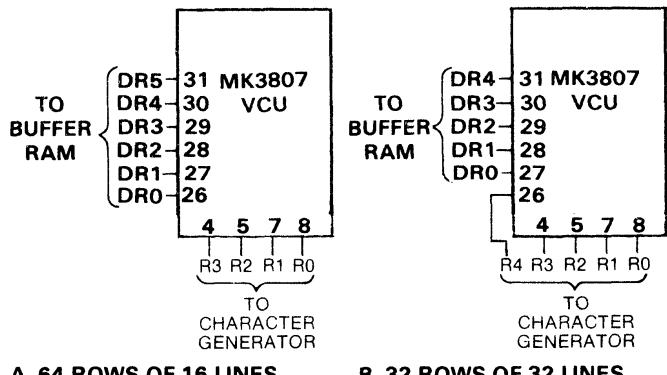
Many advantages are realized using Row-Column (X-Y) Addressing. Among these are:

Oversize Characters

Character fonts with heights greater than 16 dots (raster lines) can be achieved. This is done by using the LSB of the row counter (DR0) as the MSB of the raster scan counter (R4), and then moving the remaining bits of the row counter down one bit (DR1 becomes DR0, etc.). This is achieved by connecting the pins of the VCU in a different configuration. No additional components are required. This is shown in Figure 6. In addition, the VCU must be programmed for twice the desired number of data rows; thus using the above configuration (Figure 6), 32 rows of data with up to 32 lines per character (or 16 rows of data with up to 64 lines per character) can be accomplished.

USING THE VCU WITH CHARACTER FONTS OF HEIGHTS GREATER THAN 16 DOTS (LINES)

Figure 6



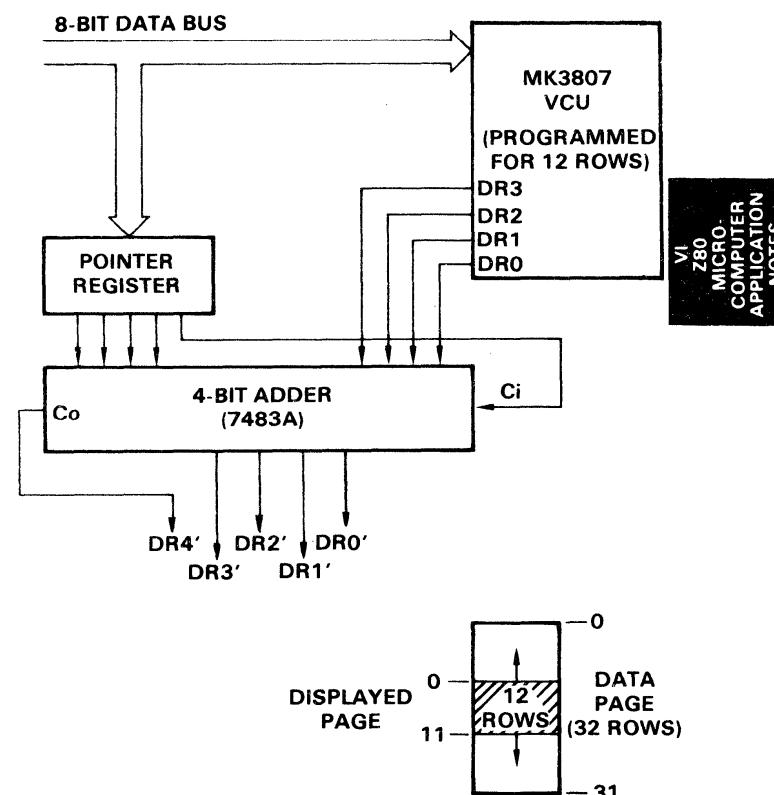
Page Scrolling

Scrolling a smaller page through a larger page (1K in 4K) can be done on a row by row basis. If the DR0-DR5 lines are offset by a pointer register, the smaller page can be moved up or down inside the larger page by the offset number of rows. This is shown in Figure 7. In this example, if the pointer register contains zero, the VCU will address the first 12 lines of the 32 line page. When the pointer register contains ten, the VCU will address rows 10 to 21. Thus, by loading the pointer register (from the microprocessor data bus), the display can scroll row by row through the data base.

Software Addressing

Most programmers use X — Y (row-column) addressing when writing software for CRT terminals. This makes it easier to blank the bottom line when scrolling, changing cursor positions, etc. Therefore, by having row-column addressing in the VCU, the address bus of the microprocessor can also have the preferred row-column addressing, and the two buses can be mapped together as shown in Figure 8. Without this feature, a software algorithm would have to convert a row-column address to binary address every time the microprocessor wanted to access the frame buffer. This algorithm usually requires a 16 bit multiplication. Thus the VCU, by utilizing row-column addressing, can save significant overhead and program execution time.

SCROLLING A 12 ROW PAGE THRU A 32 ROW PAGE
Figure 7

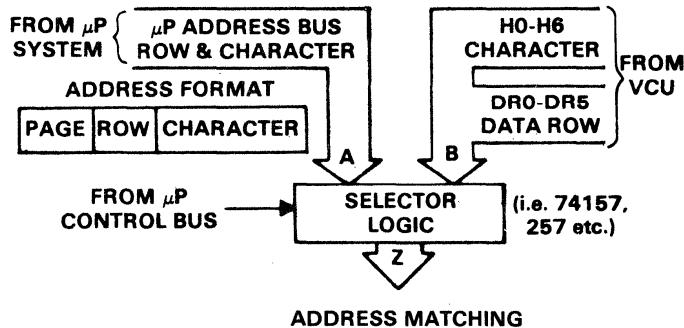


MEMORY MULTIPLEXING

The character column and character row outputs combine to form the character address bus. This bus, along with the microprocessor address bus, is connected to a 2 X 1 selector which addresses the character frame buffer RAM. Figure 8 shows the selector and the mapping for the various formats of the standard VCU. Numerous methods are available to build 2 X 1 selectors. One low-cost technique uses three

ADDRESS BUS MAPPING

Figure 8



ADDRESS BUS MAPPING

Table 1

	SELECTOR																
μP ADDRESS BUS (UNUSED BITS ARE FOR PAGE LOCATION)	INPUT (A)	AB12	AB11	AB10	AB9	AB8	AB7	AB6	AB5	AB4	AB3	AB2	AB1	AB0			
20 & 32 CHARACTERS/LINE																	
FUNCTIONS		ROW								CHARACTER							
VCU OUTPUTS	INPUT (B)		DR5	DR4	DR3	DR2	DR1	DR0		H4	H3	H2	H1	HO			
SELECTOR OUTPUTS	OUTPUT (Z)		Y5	Y4	Y3	Y2	Y1	Y0	X4	X3	X2	X1	X0				
40 & 64 CHARACTERS/LINE																	
FUNCTIONS		ROW								CHARACTER							
VCU OUTPUTS	INPUT (B)		DR5	DR4	DR3	DR2	DR1	DR0		H5	H4	H3	H2	H1	HO		
SELECTOR OUTPUTS	OUTPUT (Z)		Y5	Y4	Y3	Y2	Y1	Y0	X5	X4	X3	X2	X1	X0			
72, 80 & 96 CHARACTERS/LINE																	
FUNCTIONS		ROW								CHARACTER							
VCU OUTPUTS	INPUT (B)	DR5	DR4	DR3	DR2	DR1	DR0		H6	H5	H4	H3	H2	H1	HO		
SELECTOR OUTPUTS	OUTPUT (Z)	Y5	Y4	Y3	Y2	Y1	Y0	X6	X5	X4	X3	X2	X1	X0			
132 CHARACTERS/LINE																	
FUNCTIONS		ROW								CHARACTER							
VCU OUTPUTS	INPUT (B)	DR4	DR3	DR2	DR1	DR0	H7	H6	H5	H4	H3	H2	H1	HO			
SELECTOR OUTPUTS	OUTPUT (Z)	Y4	Y3	Y2	Y1	Y0	X7	X6	X5	X4	X3	X2	X1	X0			

74157 or equivalent (74LS157 or 257, 9322, etc.) quad 2 X 1 selector chips. Figure 8 tabulates the mapping on to the microprocessor address bus into the selector with the DR and H lines of the VCU. The output of the selector (Z), is decomposed into two fields, row (Y) and column or character (X). Refer to Table 1.

Memory Addressing

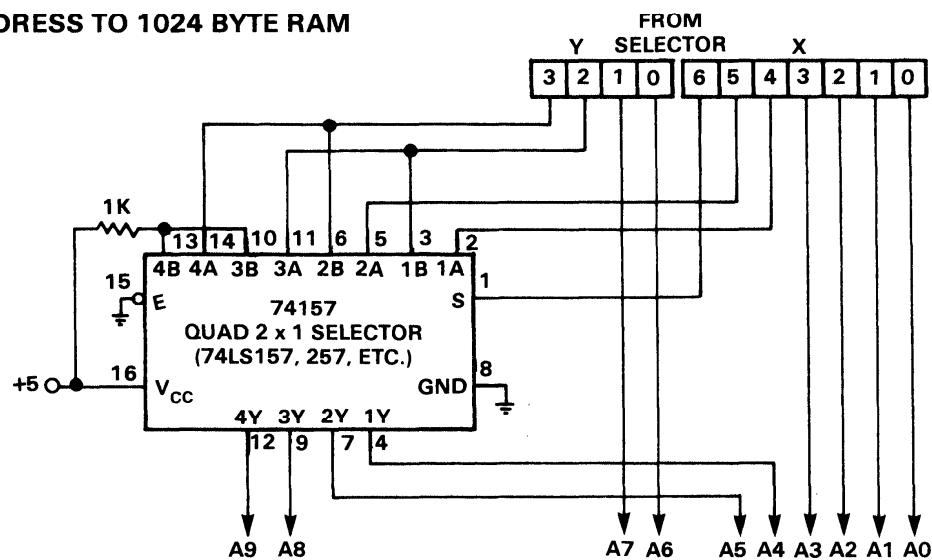
When the number of characters per row is non-binary, i.e. 80, addressing the frame buffer RAM is wasteful of memory. To solve this problem and still retain the advantages of row-column addressing, an address mapping is performed. The output of the selector (Z) is connected to another 74157 quad 2 X 1 selector chip or equivalent. Figures 6A, B, and C show the connection for 12 rows (1K), 24 rows (2K), and 48 rows (4K) of 80 characters. Figure 5 shows the mapping technique. The first 64 characters are mapped directly and the next 16 characters (H6 = 1) are mapped in a higher part of the RAM. The microprocessor address (row and column), is overlayed onto the VCU address bus (row and column) via the selector. The output of the selector maps into the frame buffer. Thus, every character is addressed by its row and column from both the microprocessor and the VCU. The

**MEMORY MAPPING CIRCUITS FOR 72
OR 80 CHARACTERS/LINE**

Figure 9

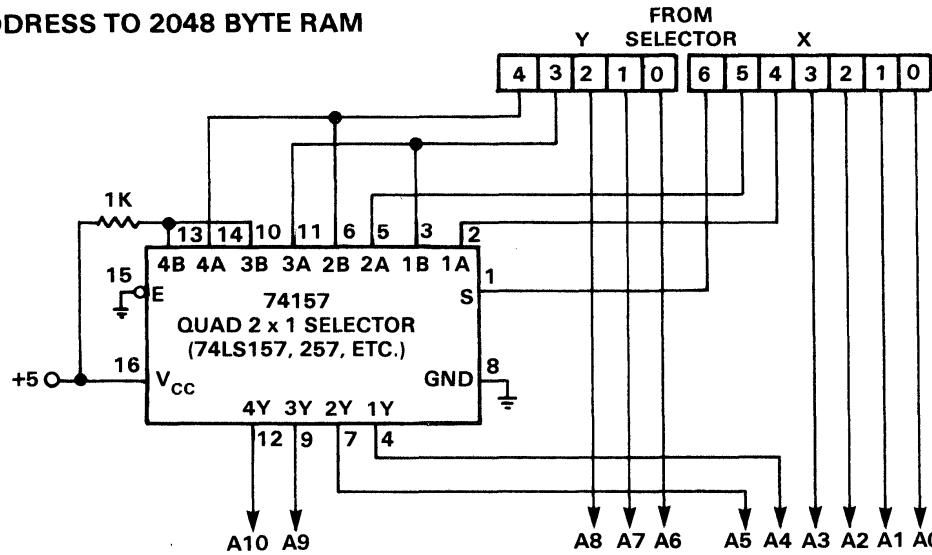
**10 BIT BINARY ADDRESS TO 1024 BYTE RAM
12 LINES INTO 1K**

Figure 9A



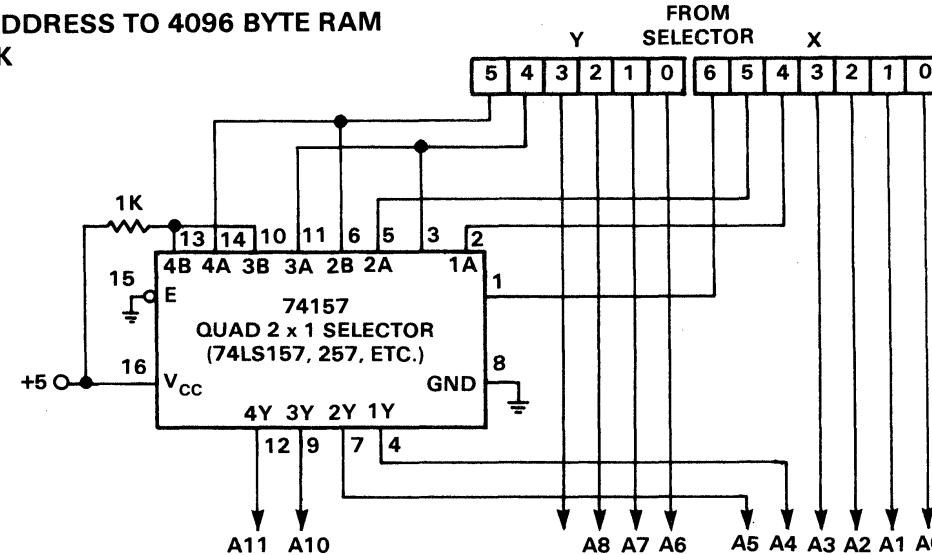
**11 BIT BINARY ADDRESS TO 2048 BYTE RAM
24 LINES INTO 2K**

Figure 9B



**12 BIT BINARY ADDRESS TO 4096 BYTE RAM
48 LINES INTO 4K**

Figure 9C



same memory location will be accessed whether the identical address originates from the microprocessor or VCU address bus.

OPERATION

The character frame buffer RAM is initially loaded via the microprocessor data and address buses (see Figure 1). After the microprocessor has loaded the character frame buffer RAM with a complete page, the selector flip-flop is switched (via the microprocessor control bus) so that the RAM is addressed by the character address bus of the VCU. In this mode the VCU operates independent of the microprocessor by addressing the character frame buffer RAM which sends the ASCII data to the CRT character generator. The selected character is then further decomposed by the raster scan counter (R0-R3), from the VCU, and loaded into the character generator shift register. This bit pattern is then serially shifted out at the video dot clock frequency and the data can be encoded so as to compose the video signal.

One possible way to change the data in the frame buffer (which is in microprocessor address space but physically separate) is: whenever the data in the character frame buffer is to be changed or updated, the microprocessor (via the control bus) sets an external flip-flop. The output of this flip-flop is ANDed with the vertical sync signal from the VCU. When this occurs an interrupt is generated to the microprocessor. This alerts the microprocessor to the fact that the vertical blanking interval has begun; it then switches the address selector (via control bus) so that the character frame buffer is now addressed by the microprocessor instead of the VCU. Since the system is in the vertical blanking interval, the screen is blank at this time. Using the American standard of $63.5 \mu s$. per horizontal line and a typical value of 21 horizontal lines for the blanking interval, this gives the system 1.33 ms. in which the microprocessor can change data in the character frame buffer. If this time is not sufficient, the 1.33 ms. window will appear every 1/60 of a second allowing the microprocessor to change part of the RAM data each time.

After the microprocessor has completed its updating of the character frame buffer RAM, it resets the external flip-flop (via the control bus) and switches the selector back to the character address bus of the VCU. Then the microprocessor goes about its normal system operation without being interrupted or having its throughput slowed down. This is because the VCU refreshes the CRT independently with the character frame buffer RAM, supplying the data, while the microprocessor operates at full speed with its own RAM and ROM. This method is more efficient for microprocessor throughput and control as opposed to having to DMA (cycle steal) or interrupt the processor continually, thereby reducing its throughput.

SYNC-LOCK

Some applications require adding alphanumeric characters (text) or graphics to the same screen as closed circuit or

external (off-the-air) video. Figure 11 illustrates a simple technique of externally synchronizing the VCU using 2 chips (7474 and 7402 or equivalent). The external video can come from a closed circuit television system, off-the-air television, or some other video display system. The technique involves stopping the character clock (DCC) when the VCU sync occurs and restarting it when the external sync occurs. In this way, the VCU will be synchronized to the external video. One requirement for the reliable operation of this system is that the VCU horizontal and vertical sync rates must be programmed to be slightly faster than the external sync rate (i.e., the horizontal line counter register of the VCU must be programmed to be less than $63.5 \mu s$, which is the American TV horizontal rate).

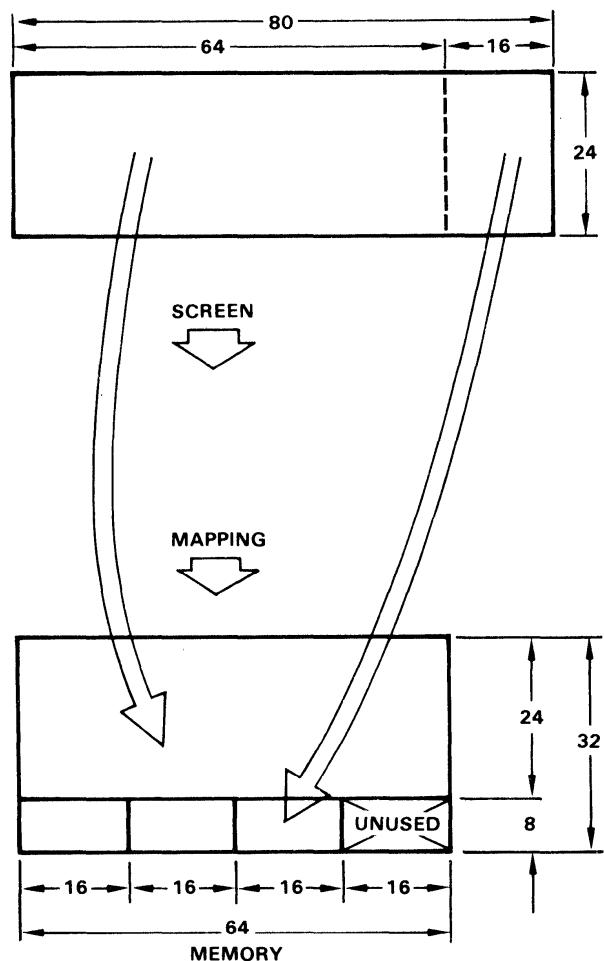
HOW TO PROGRAM THE MK3807 VCU

In order to pick the correct video dot clock frequency and to program the registers in the VCU, it is first necessary to determine several key parameters. Among these parameters are: the vertical refresh rate, the number of horizontal raster lines per frame, the number of characters per line and the format of the characters.

Tables 2A, B list work sheets which give the designer an

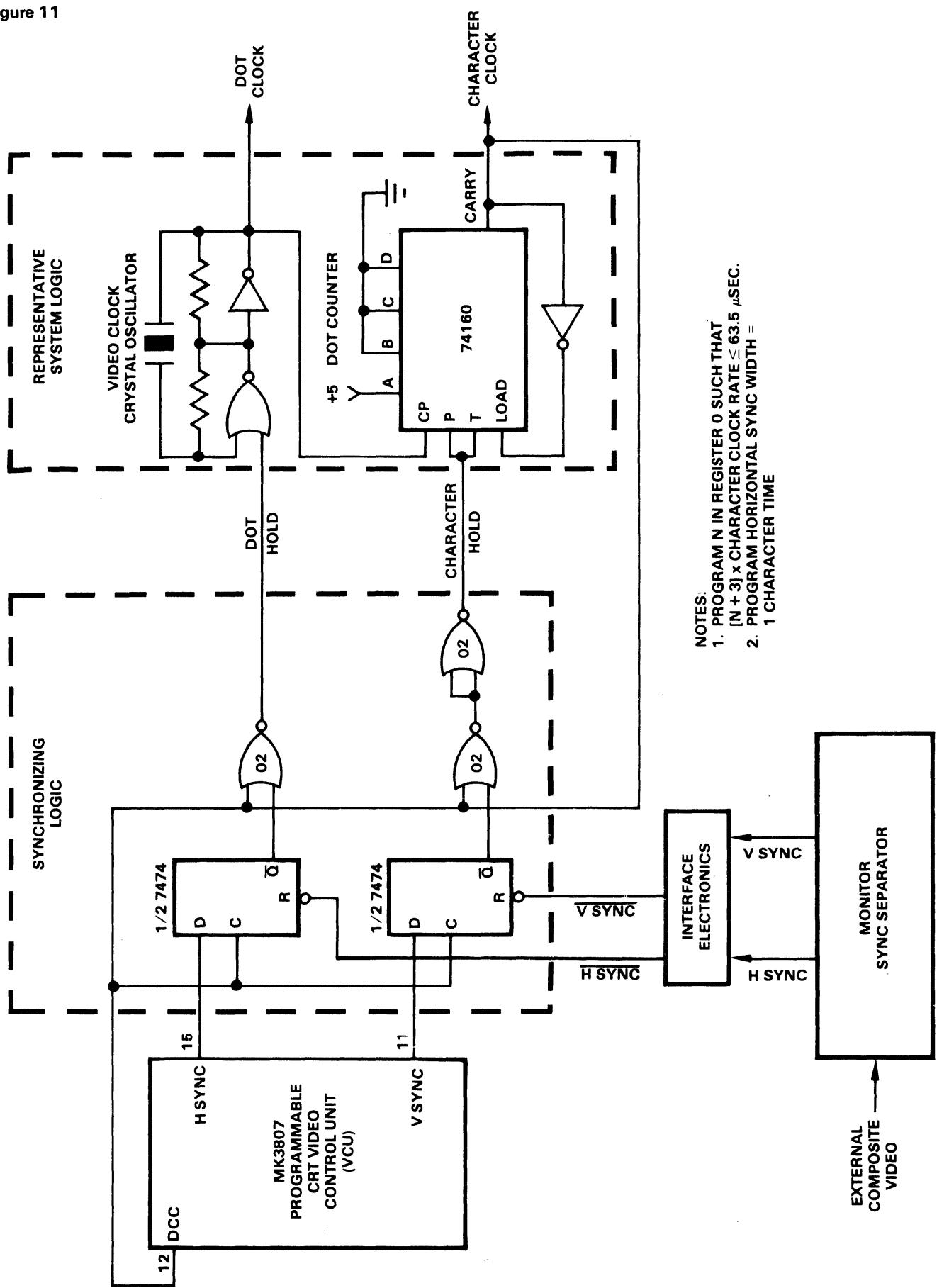
ADDRESS COMPRESSION SCHEME FOR 80 CHARACTERS/LINE

Figure 10



MK3807 EXTERNAL VIDEO SYNCHRONIZING CIRCUIT

Figure 11



orderly method of determining the frequencies and register contents from the above parameters. In order to demonstrate its use, typical examples will be shown.

EXAMPLE FOR 80 CHARACTERS BY 24 ROWS

A 7 X 9 character matrix is chosen as it is the most popular for the display of both upper and lower case characters. Also, a non-interlaced system is chosen. The character block of 9 X 12 allows for a 2 dot space between characters and a 3 line space between data rows. The impact of the character block size on the horizontal frequency and the video clock rate will be shown below. A frame refresh rate of 60Hz is chosen for this example. These numbers can be modified for 50Hz systems.

This system will have 24 rows of data and 80 characters per data row. Thus, there are (24 X 12) 288 active scan lines.

The monitor chosen for this example is capable of accepting a composite video signal or separate TTL horizontal and vertical sync pulses. The sum of the horizontal sync delay (front porch), horizontal sync pulse, and horizontal scan delay (back porch) is the horizontal blanking interval. This interval is required as a window in the horizontal scan period to allow retrace. The retrace time is internal to the CRT monitor; this time is a function of monitor horizontal scan components. This time, at a minimum, is the time it takes the display to return from the right to the left hand side of the display. The retrace time is less than the horizontal blanking interval. The horizontal blanking interval is normally about 20% of the total horizontal scanning period. See Figure 12 for horizontal and vertical timing, and Figure 13 for derived register bit assignments.

In an 80 character per data row system, this would give 20 character times for the sum of the Front Porch, Horizontal Sync Pulse, and Back Porch. In the example of table 2C, a

sum of 22 character time is used to illustrate that some flexibility exists in the choice of these parameters.

The vertical scanning frequency can be obtained by counting the total number of horizontal lines. The total number of scan lines generated for a vertical field equals the number of data rows times the number of lines per character plus the vertical sync delay plus the vertical sync pulse plus the vertical scan delay.

Vertical sync delay is the number of scan lines delay before vertical sync. Vertical sync pulse width should be expressed in scan line units. The VCU is fixed at the standard vertical sync width of 3 horizontal scan lines (3H). Scan line delay is the delay between vertical sync and the display information in scan line units. The sum of the vertical sync and the 2 delays in the vertical blanking interval is normally 5% to 8% of the total number of scan lines.

The vertical period (for 60Hz vertical refresh rate) can be calculated as: 1 divided by 60Hz = 16.67 ms.

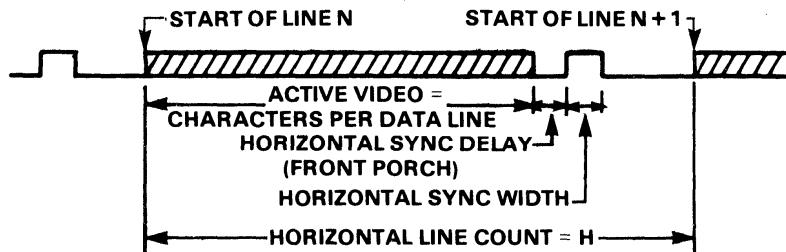
Thus, the vertical blanking period (at 8%) equals 1.3 ms. In the example of table 2C, the sum of the "Front Porch, Vertical Sync Pulse, and Back Porch" are 22 scan lines long. Again, some flexibility exists in the choice of these parameters.

Adding the displayed lines (24 X 12 = 288) plus the vertical blanking interval ($0 + 3 + 19 = 22$), 310 horizontal scan lines are required. These 310 lines must be repeated 60 times a second (every 16.67 ms.). Thus 18,600 horizontal scan lines per second is the horizontal frequency. It can now be seen that any further increase in the number of scan lines per data character block will cause a direct increase in the horizontal frequency, possibly to a point beyond the monitor's specification.

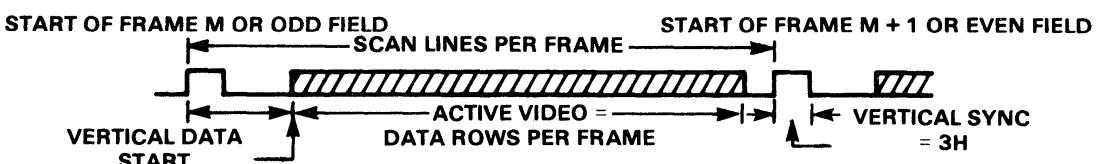
HORIZONTAL AND VERTICAL TIMING

Figure 12

HORIZONTAL TIMING



VERTICAL TIMING



MK3807 VCU WORK SHEET**Table 2A**

1. H CHARACTER MATRIX (No. of Dots):
2. V CHARACTER MATRIX (No. of Horiz. Scan Lines):
3. H CHARACTER BLOCK (Step 1 + Desired Horiz. Spacing = No. in Dots):
4. V CHARACTER BLOCK (Step 2 + Desired Vertical Spacing = No. in Horiz. Scan Lines):
5. VERTICAL FRAME (REFRESH) RATE (Freq. in Hz):
6. DESIRED NO. OF CHARACTER ROWS:
7. TOTAL NO. OF ACTIVE "VIDEO DISPLAY" SCAN LINES
(Step 4 x Step 6 = No. in Horiz. Scan Lines):
8. VERT. SYNC DELAY (No. in Horiz. Scan Lines):
9. VERT. SYNC (No. in Horiz. Scan Lines; $T = \underline{\hspace{2cm}} \mu s^*$):
10. VERT. SCAN DELAY (No. in Horiz. Scan Lines; $T = \underline{\hspace{2cm}} ms^*$):
11. TOTAL VERTICAL FRAME (Add steps 7 thru 10 = No. in Horiz. Scan Lines):
12. HORIZONTAL SCAN LINE RATE (Step 5 x step 11 = Freq. in KHz):
13. DESIRED NO. OF CHARACTERS PER HORIZ. ROW:
14. HORIZ. SYNC DELAY (No. in Character Time Units; $T = \underline{\hspace{2cm}} \mu s^{**}$):
15. HORIZ. SYNC (No. in Character Time Units; $T = \underline{\hspace{2cm}} \mu s^{**}$):
16. HORIZ. SCAN DELAY (No. in Character Time Units; $T = \underline{\hspace{2cm}} \mu s^{**}$):
17. TOTAL CHARACTER TIME UNITS IN (1) HORIZ. SCAN LINE
(Add Steps 13 thru 16):
18. CHARACTER RATE (Step 12 x Step 17 = Freq. in MHz):
19. CLOCK (DOT) RATE (Step 3 x Step 18 = Freq. in MHz):

*Vertical Interval

**Horizontal Interval

VI
Z80
MICRO
COMPUTER
APPLICATION
NOTES

MK3807 VCU WORK SHEET
Table 2B

REG. #	ADDRESS A3—A0	FUNCTION	BIT ASSIGNMENT	HEX.	DEC.
0	0000	HORIZ. LINE COUNT _____	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	—	—
1	0001	INTERLACE _____ H SYNC WIDTH _____ H SYNC DELAY _____	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	—	—
2	0010	SCANS/DATA ROW _____ CHARACTERS/ROW _____	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> X	—	—
3	0011	SKEW CHARACTERS _____ DATA ROWS _____	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	—	—
4	0100	SCANS/FRAME _____ X = _____	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	—	—
5	0101	VERTICAL DATA START = 3 + VERTICAL SCAN DELAY: SCAN DELAY _____ DATA START _____	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	—	—
6	0110	LAST DISPLAYED DATA ROW (= DATA ROWS)	<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	—	—

MK3807 VCU WORK SHEET

Table 2C

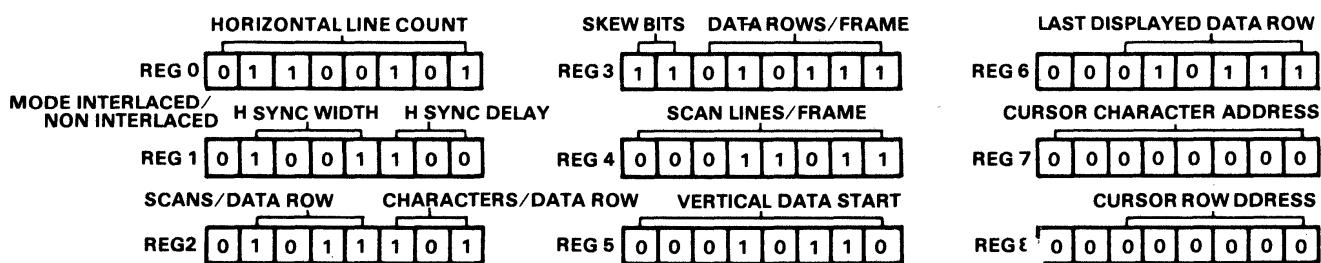
1. H CHARACTER MATRIX (No. of Dots):	7
2. V CHARACTER MATRIX (No. of Horiz. Scan Lines):	9
3. H CHARACTER BLOCK (Step 1 + Desired Horiz. Spacing = No. in Dots):	9
4. V CHARACTER BLOCK (Step 2 + Desired Vertical Spacing = No. in Horiz. Scan Lines):	12
5. VERTICAL FRAME (REFRESH) RATE (Freq. in Hz):	60
6. DESIRED NO. OF CHARACTER ROWS:	24
7. TOTAL NO. OF ACTIVE "VIDEO DISPLAY SCAN LINES" (Step 4 x Step 6 = No. in Horiz. Scan Lines):	288
8. VERT. SYNC DELAY (No. in Horiz. Scan Lines):	0
9. VERT. SYNC (No. in Horiz. Scan Lines; T = <u>161.29</u> μ s*):	3
10. VERT. SCAN DELAY (No. in Horiz. Scan Lines; T = <u>1.02</u> ms*):	19
11. TOTAL VERTICAL FRAME (Add steps 7 thru 10 = No. in Horiz. Scan Lines):	310
12. HORIZONTAL SCAN LINE RATE (Step 5 x step 11 = Freq. in KHz):	18.6
13. DESIRED NO. OF CHARACTERS PER HORIZ. ROW:	80
14. HORIZ. SYNC DELAY (No. in Character Time Units; T = <u>2.11</u> μ s**):	4
15. HORIZ. SYNC (No. in Character Time Units; T = <u>4.74</u> μ s**):	9
16. HORIZ. SCAN DELAY (No. in Character Time Units; T = <u>4.74</u> μ s**):	9
17. TOTAL CHARACTER TIME UNITS IN (1) HORIZ. SCAN LINE (Add Steps 13 thru 16):	102
18. CHARACTER RATE (Step 12 x Step 17 = Freq. in MHz):	1.8972
19. CLOCK (DOT) RATE (Step 3 x Step 18 = Freq. in MHz):	17.0748

*Vertical Interval

**Horizontal Interval

BIT ASSIGNMENT

Figure 13



XTAL Frequency

At a frequency of 18.6KHz a scan line takes $53.76 \mu s$. In this time 102 characters (80 displayed + 22 blanked) have to be accessed. Thus the character time is 527.06 ns ($53.76 \mu s / 102$). Since each character is 9 dots in this example (7 character and 2 blank), the dot period is 58.56 ns ($527.06 \text{ ns} / 9$). The inverse of the dot period is the video dot clock XTAL frequency. For this example, the video dot clock XTAL is $1 / 58.56 \text{ ns} = 17.0748 \text{ MHz}$ ($53.76 \mu s / 102$). Since each character is 9 dots in this example (7 character and 2 blank), the dot period increases in the video clock rate, possibly to a point beyond the monitor's specification.

A more detailed example, using 40 character by 12 row format, follows.

Having chosen the display format and display monitor, the actual settings for the VCU registers can now be established. See Table 2C.

EXAMPLE FOR 40 CHARACTER BY 12 ROWS

Using the VCU worksheet (Table 2A), steps 1 and 2 determine the character matrix. In this example, a 7×9 dot matrix will be used, thus in step 1, 7 dots are used horizontally and in step 2, 9 scan lines are used vertically. This defines the character size (other character sizes might be 5×7 etc.). Steps 3 and 4 determine the character block size. The character block is composed of the character matrix along with both the horizontal and vertical blank

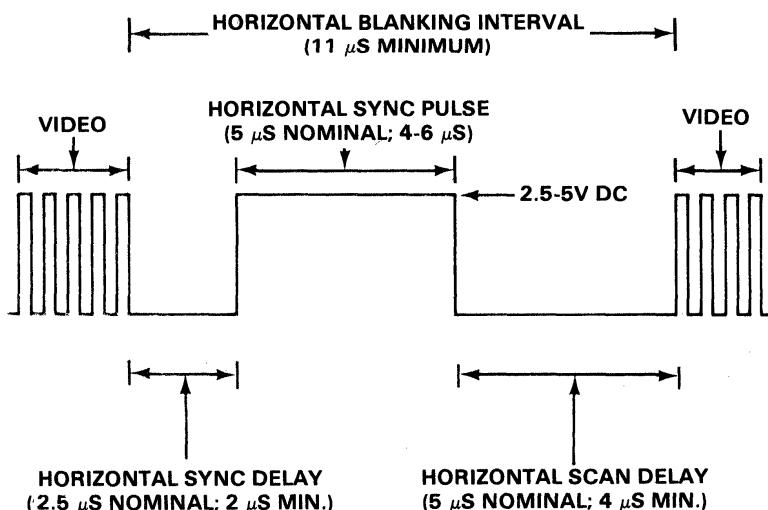
spaces between characters. Step 3 shows the H character block for this example to be 7 dots from step 1 plus 2 additional dots for blank space, giving a total of 9. Step 4 shows the vertical height (V character block) being 9 lines from step 2, plus 3 additional raster lines for vertical spacing, giving a total of 12. The next parameter is the vertical frame refresh rate and this example uses the American Standard of 60Hz (in this example the non-interlace mode will also be used).

As this example uses twelve rows of data, step 6 indicates 12. Step 7 determines the number of active video display raster scan lines. This is determined by taking the number of raster scan lines from step 4 and multiplying that by the number of data rows in step 6, thus giving us the number of displayed horizontal scan lines. In this example, multiply 12 raster lines per data row by 12 data rows to give 144 active video raster scan lines.

The next portion of this example is dependent upon the characteristics of the video monitor being used. For the purposes of this example a standard sync driven video monitor using RS-170 non-interlace sync is used. In accordance with the standard for this monitor, the vertical sync pulse width will be between 180 and $200 \mu s$, with $190 \mu s$ as the nominal value. In addition, the vertical blanking interval, which is made up of the vertical sync pulse and the 2 delays, is defined as being 1 ms. minimum. The same monitor specification defines the horizontal sync pulse width as being between 4 and $6 \mu s$, with $5 \mu s$ as the nominal horizontal sync pulse width. In addition, the horizontal sync delay or front porch is defined as $2.5 \mu s$.

MONITOR HORIZONTAL TIMING

Figure 14



nominally with a 2 μ s. minimum. At the same time, the horizontal blanking interval, which is composed of the front porch, horizontal sync pulse, and the back porch is defined as 11 μ s. minimum. See Figures 14 and 15.

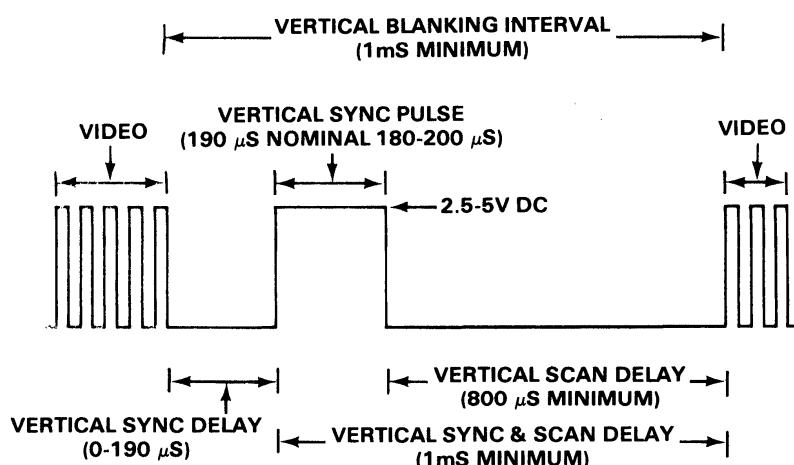
The monitor characteristics determine the values for steps 9 and 10. Step 9 lists the vertical sync pulse width. The VCU has a fixed vertical sync pulse width of 3 horizontal raster scan lines (3H). Later, the period of a horizontal raster scan line will be determined and verified that this meets the RS-170 specification. Enough time must be allowed for vertical retrace and some blanking at the top of the screen. This is indicated in step 10 as the vertical scan delay. The VCU can be programmed for a vertical scan delay between 0 and 255 raster scan lines to allow utilization of various types of monitors, as well as to position the data vertically on the screen. For purposes of this example, a vertical scan delay of 19 raster lines is chosen. After the horizontal period is determined, it can be verified that these values comply with the specification. Step 11 is the total number of raster lines per frame or, in other words, the number of raster lines per vertical refresh time. Normally, this will be determined by adding to the number of displayed scan lines, the vertical sync pulse width, the vertical scan delay, and the vertical sync delay which has not yet been determined. However, in this case, since the example uses a standard monitor, it is possible to work backwards. Therefore, for step 11 we will enter 262 raster lines per frame (a typical number of raster lines/field of a standard monitor). Now work backwards to step 8 and determine the vertical sync delay. This is the number of raster lines between the last displayed video raster line and the beginning of vertical sync. Subtracting

144, 19, and 3 from 262 leaves 96, thus for step 8, 96 horizontal lines is the vertical sync delay. We have now determined the vertical timing waveform for this example. The next part of the example is to determine the horizontal scan line rate or how many raster lines per second will be displayed. This is determined by multiplying the vertical frame refresh rate from step 5; in this case 60 frames per second by the total number of raster lines per frame from step 11, in this case 262. The product will be 15,720 raster lines per second. This is the horizontal scan rate. The horizontal period is determined by taking the inverse of horizontal scan rate, 1 divided by 15,720 Hz is 63.6132 μ s. This is the time of 1 horizontal raster line. This information is now used to go back and check on meeting the specifications in steps 9 and 10. Step 9 lists 3 horizontal lines as the vertical sync pulse width. $3 \times 63.6132 \mu$ s. yields 190.84 μ s. This is the nominal value specified for the monitor. Step 10 lists the vertical scan delay as 19 raster lines multiplying that by 63.61 μ s. yields 1.21 ms., thus the values picked for the above parameters meet the specification for the monitor.

In step 13 the desired number of active display characters per horizontal data row is listed. 40 character per row have been chosen. Steps 14, 15 and 16 are now selected using the horizontal period and the monitor specifications. Step 14 is the horizontal sync delay or front porch. In this case 2 character times. The period of a character will be determined later in this example which will be used to verify that this parameter meets the RS-170 specification given earlier. In step 15 the horizontal sync width is chosen to be 4 character times and in step 16 the horizontal scan delay is

MONITOR VERTICAL TIMING

Figure 15



chosen to also be 4 character times. Step 17 is the total number of character times per horizontal scan line and this is determined by adding steps 13 through 16, thus we add $40+2+4+4=50$ character times per horizontal scan line. In step 18 the character rate is determined by multiplying the horizontal line rate of step 12 by the total character units per horizontal line, thus, $15,720 \times 50 = 786,000$ characters per second. The character period is the inverse of the character rate, thus 1 over 786,000 yields a character period of 1.272 μs . This information is used to verify steps 14, 15, and 16. In step 14 the horizontal sync delay was chosen as 2 character units. 2 times 1.272 μs yields 2.54 μs . Step 15, the horizontal sync width was 4 character units. 4 times 1.272 μs yields 5.089 μs . and similarly, step 16, four character units also is 5.089 μs . These three values are in agreement with the specification for the monitor. The next step is to determine the video dot clock frequency. It is determined by multiplying the number of dots per character from step 3 by the character rate in step 18, $9 \times 786 \text{ KHz} = 7.074 \text{ MHz}$. Thus, the crystal frequency required for this example is 7.074 MHz and the dot clock counter divisor N is 9 (from step 3).

Register Programming

Register 0 (Horizontal Line Count) determines the total number of character units per horizontal line. From step 17 we have determined that there would be 50 character units

per line. This register is loaded with (N — 1) the decimal number 49.

Register 1 contains 3 fields. The first field is the most significant bit and this determines the interlaced or non-interlaced mode of operation. This example uses the non-interlaced mode, therefore, bit 7 is loaded with a 0. The next field is the horizontal sync pulse width and this field is bits 6 through 3. Step 15 determines that the horizontal sync width is 4 character times. Therefore the binary equivalent of 4 is loaded into these bits. Thus bits 6 through 3 are loaded with 0100. The third field is the horizontal sync delay, step 14 determines that this is 2 character time units. Therefore, bits 2 through 0 are loaded with 010.

Register 2 contains 2 fields, with the most significant bit unused. Bits 6 through 3 determine the scans per data row. In this example from step 4, there will be 12 raster lines per data row, and from the VCU data sheet note this is an N + 1 register. Therefore the decimal number eleven is loaded into bits 6 through 3. the second field is characters per data row, bits 2 through 0. In this example 40 active characters per data row was chosen. The VCU data sheet specifies that 010 in this field will give 40 characters per data row, thus bits 2 through 0 are loaded with 010.

Register 3 also contains 2 fields. The first field, bits 7 and 6, are the skew bits. These bits allow the hardware designer to

MK3807 VCU WORK SHEET

1. H CHARACTER MATRIX (No. of Dots):	7
2. V CHARACTER MATRIX (No. of Horiz. Scan Lines):	9
3. H CHARACTER BLOCK (Step 1 + Desired Horiz. Spacing = No. in Dots):	9
4. V CHARACTER BLOCK (Step 2 + Desired Vertical Spacing = No. in Horiz. Scan Lines):	12
5. VERTICAL FRAME (REFRESH) RATE (Freq. in Hz):	60
6. DESIRED NO. OF CHARACTER ROWS:	12
7. TOTAL NO. OF ACTIVE "VIDEO DISPLAY SCAN LINES" (Step 4 x Step 6 = No. in Horiz. Scan Lines):	144
8. VERT. SYNC DELAY (No. in Horiz. Scan Lines):	9/6
9. VERT. SYNC (No. in Horiz. Scan Lines; T = <u>190.84</u> μs^*):	3
10. VERT. SCAN DELAY (No. in Horiz. Scan Lines; T = <u>1.21</u> ms*):	19
11. TOTAL VERTICAL FRAME (Add steps 7 thru 10 = No. in Horiz. Scan Lines):	262
12. HORIZONTAL SCAN LINE RATE (Step 5 x step 11 = Freq. in KHz):	15.72
13. DESIRED NO. OF CHARACTERS PER HORIZ. ROW:	40
14. HORIZ. SYNC DELAY (No. in Character Time Units; T = <u>2.54</u> μs^{**}):	2
15. HORIZ. SYNC (No. in Character Time Units; T = <u>5.09</u> μs^{**}):	4
16. HORIZ. SCAN DELAY (No. in Character Time Units; T = <u>5.09</u> μs^{**}):	4
17. TOTAL CHARACTER TIME UNITS IN (1) HORIZ. SCAN LINE (Add Steps 13 thru 16):	50
18. CHARACTER RATE (Step 12 x Step 17 = Freq. in MHz):786
19. CLOCK (DOT) RATE (Step 3 x Step 18 = Freq. in MHz):	7.074

*Vertical Interval

**Horizontal Interval

use a slower buffer RAM memory and allow compensation for slower character generator access times. In the example shown as well as most typical applications, these bits are set for 2 character time delays, therefore bit 7 and bit 6 will both contain a 1. The other field is data rows per frame, bits 5 through 0. In Step 6 there are 12 data rows per frame, and the VCU data sheet specifies that this is an N + 1 register. Thus the decimal number eleven is loaded in bits 5 through 0.

Register 4 determines the number of horizontal raster lines per frame. From this example, step 11, specifies there are 262 raster lines per frame. The VCU data sheet specifies that there are two modes of loading this register. In the non-interlace mode (this example) the equation $2X + 256$ is equal to 262. Thus, X is equal to 3. The decimal number 3 is loaded into register 4.

Register 5 is the vertical start of data. From steps 9 and 10 in the example the vertical data start is 22 raster lines, thus the decimal number 22 is loaded into register 5.

Register 6 is the last displayed data row. This register is used for multi-line scrolling and for initialization purposes is set to the same data as in register 3, the data rows per frame. Thus, the decimal number eleven is loaded into register 6.

The following will illustrate the use of register 6 for multi-line scrolling:

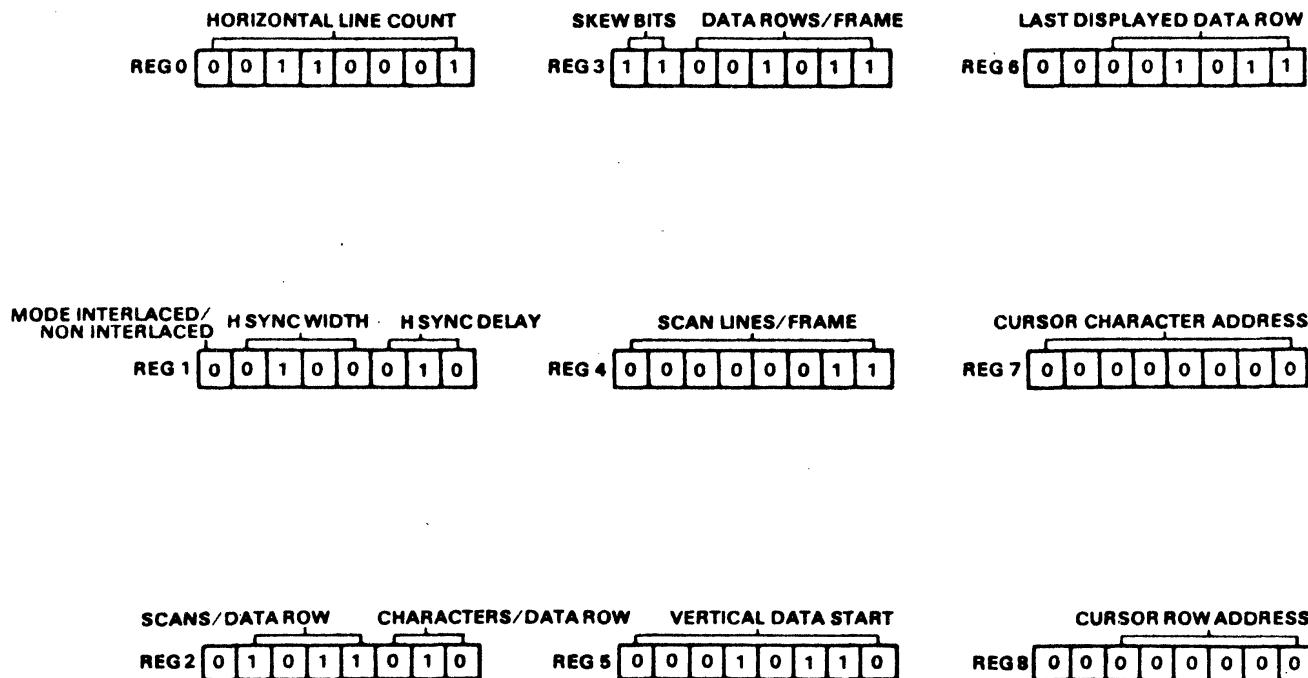
Using 12 rows of data with row 0 on top of the screen and row 11 on the bottom and as programmed in register 6 with eleven, this will be the case. Now, if another number is programmed into register 6, such as 5, data row 5 will be on the bottom of the screen, while data row 6 will be on the top followed by data row 7, 8, through to 11, followed by row 0 through 5.

Register 7 is the cursor character address. It is initialized to 0, thus it is now set to the beginning of the data row.

Register 8 is also initialized to 0. This is the cursor row address and is set to the top data row. The 2 cursor addresses (X-Y) coincide at the upper left hand corner of the screen. See the VCU work sheet on page 16.

The above is only a typical example of how to determine the frequencies, program the frequencies, and program the registers of the VCU. This is shown for illustrative purposes only and designers/programmers should determine these values for their specific CRT requirements.

BIT ASSIGNMENT CHART



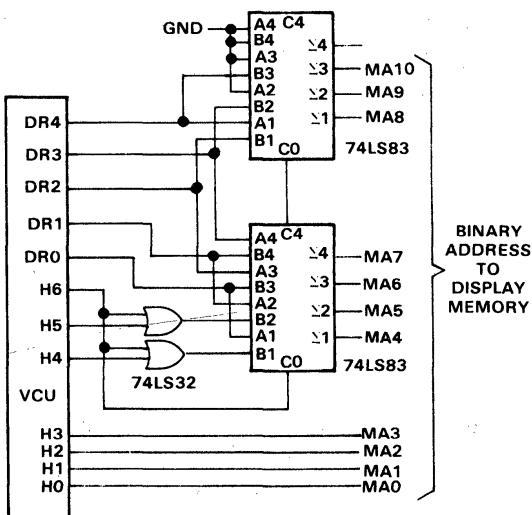
APPLICATION NOTES

Conversion of Row Column to Binary Address

With only slightly more complicated circuitry than required by memory mapping, the row column addressing outputs of the VCU may be readily changed to binary address outputs. For data formats that use 48 or 80 visible characters per data row, this can be done by the addition of two 74LS83's and a 74LS32 (or equivalent) in some formats or by the addition of the one 256x8 PROM. Figure 16 below shows the implementation for an 80 character by 24 data row display using the adders. Figure 17 is an implementation using a bipolar PROM.

80x24 DISPLAY WITH BINARY ADDRESS USING 74LS83 ADDERS

Figure 16

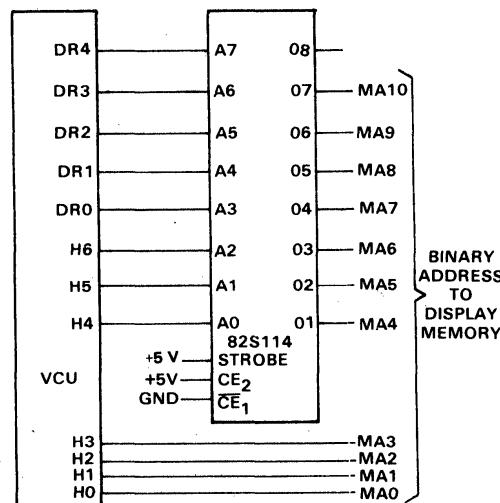


In essence the adders are used to add groups of 16. Since there are 5 groups of 16 in each data row of 80 characters, the adders effectively multiply the data row count (DR0-DR4) by 5 to obtain the starting binary address for each row. This is done by adding DR0-DR4 to itself shifted two positions to the left. Within each data row, H6, H5, and H4 are used to add from 0 to 4 groups of 16. The PROM configuration is merely a table look-up implementation of the adder configuration.

The PROM configuration can be programmed to provide binary addresses for any number of groups of 16 characters per data row (i.e., 48, 80, 96, 112, 144, 160). Table 3 shows some typical mapping for an 80x24 display.

80x24 DISPLAY WITH BINARY ADDRESS USING 256x8 PROM

Figure 17



TYPICAL MAPPING OF 80x24 DISPLAY

Table 3

ADDRESS TABLE

D R	D R	D R	D R	D R	H6	H5	H4	H3	H2	H1	HO	ROW	COL	M A 10	M A 9	M A 8	M A 7	M A 6	M A 5	M A 4	M A 3	M A 2	M A 1	M A 0	ADDR. (BIN)
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	1	0	0	0	0	0	16	0	0	0	0	0	0	1	0	0	0	0	
0	0	0	0	0	1	0	0	1	1	1	1	0	79	0	0	0	0	1	0	0	1	1	1	1	
0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	80	
0	0	0	0	1	1	0	0	1	1	1	1	1	79	0	0	0	0	1	0	0	1	1	1	1	
0	0	0	1	0	0	0	0	0	0	0	0	2	0	0	0	0	1	0	1	0	0	0	0	160	
0	0	0	1	0	1	0	0	1	1	1	1	2	79	0	0	0	0	1	1	1	0	1	1	1	
0	0	0	1	1	0	0	0	0	0	0	0	3	0	0	0	0	1	1	1	1	0	0	0	240	
1	0	1	1	1	0	0	0	0	0	0	0	23	0	1	1	1	0	0	1	1	0	0	0	0	
1	0	1	1	1	1	0	0	1	1	1	1	23	79	1	1	1	0	1	1	1	1	1	1	1	
																								1919	

USING THE VCU FOR A 256 X 256 DOT GRAPHIC DISPLAY

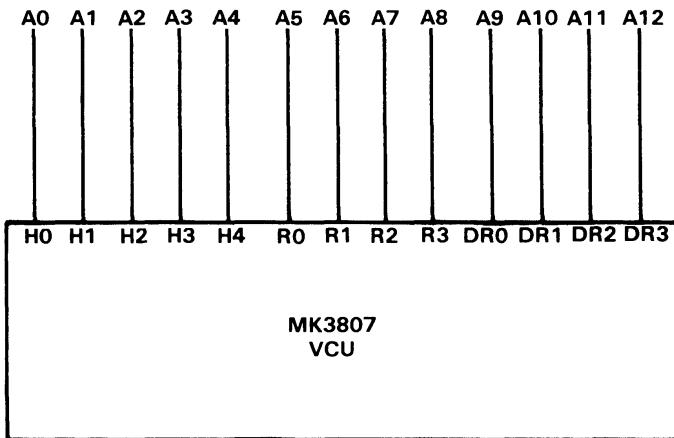
The VCU can be used for dot matrix graphic displays as well as alphanumeric displays. The following is an example of a 256 x 256 dot matrix graphic display using the raster line counter outputs (R0-R3) as part of the RAM addressing.

For this example the character width (the dot counter divisor) should be 8 dots. The VCU should be programmed (See Figure 18) for:

Characters per data row = 32
Scans per data row = 16
Data rows per frame = 16

USING THE VCU FOR A 256 x 256 DOT GRAPHIC DISPLAY

Figure 18



USING THE VCU FOR MORE THAN 128 CHARACTERS PER ROW AND MORE THAN 32 ROWS

Due to pin limitation, the most significant character count output of the VCU is multiplexed with the most significant bit of the data row counter. When the horizontal line count is greater than 128, this output (H7/DR5) automatically becomes H7. On the surface, this creates a limitation of no more than 32 data rows.

In actual fact, the row column addressing of the VCU permits the display of more than 128 characters per row and more than 32 rows per frame with only two inverters and one D-type flip flop. In the following example, the display format will be 132 characters per row by 35 data rows.

The horizontal row address will appear on outputs H0 to H7. Data row outputs DR0 to DR4 will provide five of the six bits required for the data row addressing. The circuit shown in Figure 19 will generate the required sixth row address bit.

There are many other applications of the VCU other than the alphanumeric CRT terminal as shown above.

Because of the speed and flexibility of the device, it can be used to generate television pictures (with gray scale and color), facsimile, slow-scan TV, frame storage, scan conversion, etc. Since the VCU generates composite sync (with serrations), the serial video can be combined with the composite sync to produce composite video (RS-170).

USING THE VCU FOR MORE THAN 128 CHARACTERS PER ROW AND MORE THAN 32 ROWS

Figure 19

