



## 8275H PROGRAMMABLE CRT CONTROLLER

- Programmable Screen and Character Format
- 6 Independent Visual Field Attributes
- 11 Visual Character Attributes (Graphic Capability)
- Cursor Control (4 Types)
- Light Pen Detection and Registers
- MCS-51®, MCS-85®, iAPX 86, and iAPX 88 Compatible
- Dual Row Buffers
- Programmable DMA Burst Mode
- Single +5V Supply
- High Performance HMOS-II

The Intel® 8275H Programmable CRT Controller is a single chip device to interface CRT raster scan displays with Intel® microcomputer systems. It is manufactured on Intel's advanced HMOS-II process. Its primary function is to refresh the display by buffering the information from main memory and keeping track of the display position of the screen. The flexibility designed in the 8275H will allow simple interface to almost any raster scan CRT display with a minimum of external hardware and software overhead.

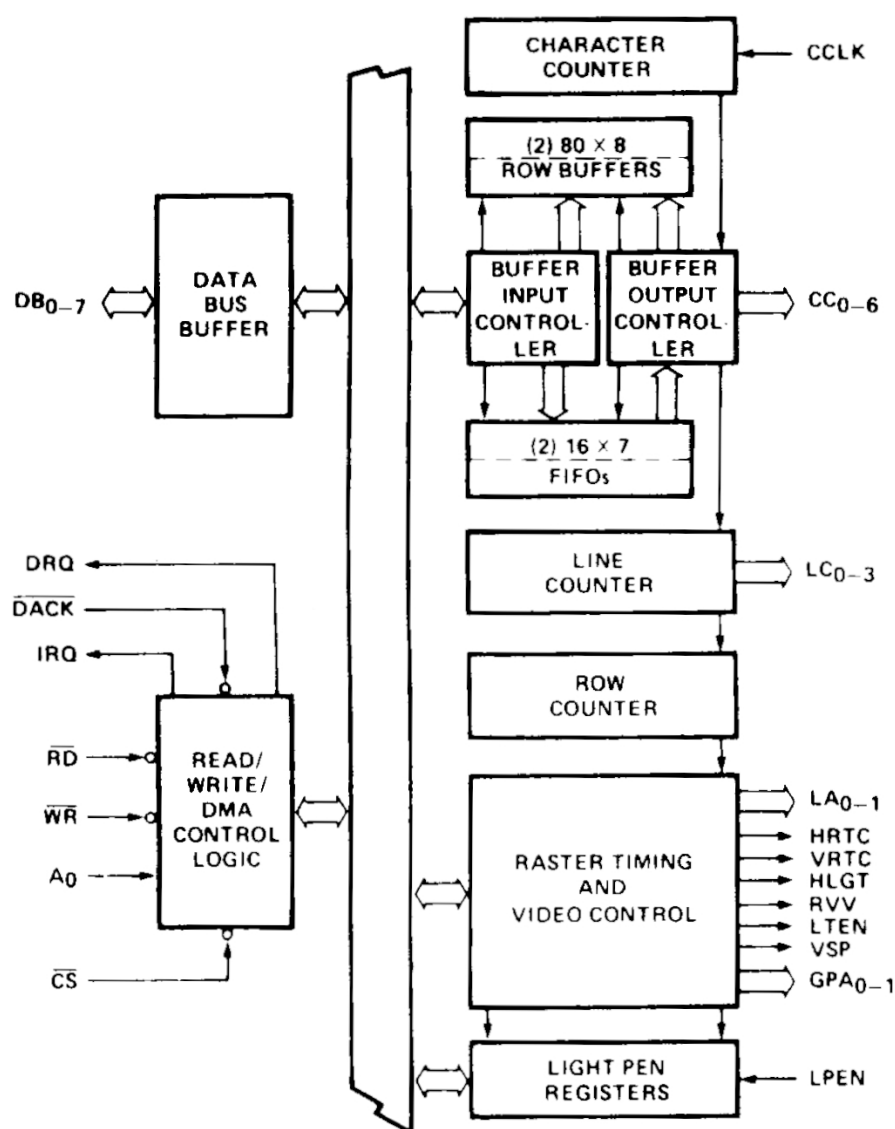
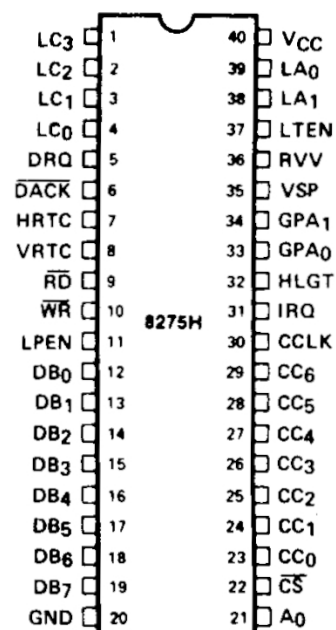


Figure 1. Block Diagram



210464-2

Figure 2. Pin Configuration

**Table 1. Pin Descriptions**

| <b>Symbol</b>  | <b>Pin No.</b>                               | <b>Type</b> | <b>Name and Function</b>  |
|--|--|-------------|---|
| LC <sub>3</sub><br>LC <sub>2</sub><br>LC <sub>1</sub><br>LC <sub>0</sub>   | 1<br>2<br>3<br>4                             | O           | <b>LINE COUNT:</b> Output from the line counter which is used to address the character generator for the line positions on the screen.  |
| DRQ  | 5  | O           | <b>DMA REQUEST:</b> Output signal to the 8257 DMA controller requesting a DMA cycle.  |
| $\overline{\text{DACK}}$   | 6  | I           | <b>DMA ACKNOWLEDGE:</b> Input signal from the 8257H DMA controller acknowledging that the requested DMA cycle has been granted.   |
| HRTC   | 7  | O           | <b>HORIZONTAL RETRACE:</b> Output signal which is active during the programmed horizontal retrace interval. During this period the VSP output is high and the LTEN output is low.   |
| VRTC   | 8  | O           | <b>VERTICAL RETRACE:</b> Output signal which is active during the programmed vertical retrace interval. During this period the VSP output is high and the LTEN output is low.   |
| $\overline{\text{RD}}$   | 9  | I           | <b>READ INPUT:</b> A control signal to read registers.  |
| $\overline{\text{WR}}$   | 10   | I           | <b>WRITE INPUT:</b> A control signal to write commands into the control registers or write data into the row buffers during a DMA cycle.  |
| LPEN   | 11   | I           | <b>LIGHT PEN:</b> Input signal from the CRT system signifying that a light pen signal has been detected.  |
| DB <sub>0</sub><br>DB <sub>1</sub><br>DB <sub>2</sub><br>DB <sub>3</sub><br>DB <sub>4</sub><br>DB <sub>5</sub><br>DB <sub>6</sub><br>DB <sub>7</sub> | 12<br>13<br>14<br>15<br>16<br>17<br>18<br>19 | I/O         | <b>BI-DIRECTIONAL THREE-STATE DATA BUS LINES:</b> The outputs are enabled during a read of the C or P ports.  |
| Ground   | 20   |             | <b>GROUND.</b>  |
| V <sub>CC</sub>  | 40   |             | <b>+ 5V POWER SUPPLY.</b>   |
| LA <sub>0</sub><br>LA <sub>1</sub>   | 39<br>38                                     | O           | <b>LINE ATTRIBUTE CODES:</b> These attribute codes have to be decoded externally by the dot/timing logic to generate the horizontal and vertical line combinations for the graphic displays specified by the character attribute codes. |

**Table 1. Pin Descriptions (Continued)**

| <b>Symbol</b>   | <b>Pin No.</b>                         | <b>Type</b> | <b>Name and Function</b>   |
|---|--|-------------|--|
| LTEN  | 37                                     | O           | <b>LIGHT ENABLE:</b> Output signal used to enable the video signal to the CRT. This output is active at the programmed underline cursor position, and at positions specified by attribute codes.   |
| RVV   | 36                                     | O           | <b>REVERSE VIDEO:</b> Output signal used to indicate the CRT circuitry to reverse the video signal. This output is active at the cursor position if a reverse video block cursor is programmed or at the positions specified by the field attribute codes.   |
| VSP   | 35                                     | O           | <b>VIDEO SUPPRESSION:</b> Output signal used to blank the video signal to the CRT. This output is active: <ul style="list-style-type: none"> <li>• during the horizontal and vertical retrace intervals.</li> <li>• at the top and bottom lines of rows if underline is programmed to be number 8 or greater.</li> <li>• when an end of row or end of screen code is detected.</li> <li>• when a DMA underrun occurs.</li> <li>• at regular intervals (<math>1/16</math> frame frequency for cursor, <math>1/32</math> frame frequency for character and field attributes)—to create blinking displays as specified by cursor, character attribute, or field attribute programming.</li> </ul> |
| GPA <sub>1</sub><br>GPA <sub>0</sub>  | 34<br>33                               | O           | <b>GENERAL PURPOSE ATTRIBUTE CODES:</b> Outputs which are enabled by the general purpose field attribute codes.  |
| HLGT  | 32                                     | O           | <b>HIGHLIGHT:</b> Output signal used to intensify the display at particular positions on the screen as specified by the character attribute codes or field attribute codes.  |
| IRQ   | 31                                     | O           | <b>INTERRUPT REQUEST.</b>  |
| CCLK  | 30                                     | I           | <b>CHARACTER CLOCK (from Dot/Timing Logic).</b>  |
| CC <sub>6</sub><br>CC <sub>5</sub><br>CC <sub>4</sub><br>CC <sub>3</sub><br>CC <sub>2</sub><br>CC <sub>1</sub><br>CC <sub>0</sub> | 29<br>28<br>27<br>26<br>25<br>24<br>23 | O           | <b>CHARACTER CODES:</b> Output from the row buffers used for character selection in the character generator.   |
| CS  | 22                                     | I           | <b>CHIP SELECT:</b> The read and write are enabled by CS.  |
| A <sub>0</sub>  | 21                                     | I           | <b>PORT ADDRESS:</b> A high input on A <sub>0</sub> selects the "C" port or command registers and a low input selects the "P" port or parameter registers.   |

## FUNCTIONAL DESCRIPTION

### Data Bus Buffer

This 3-state, bidirectional, 8-bit buffer is used to interface the 8275H to the system Data Bus.

This functional block accepts inputs from the System Control Bus and generates control signals for overall device operation. It contains the Command, Parameter, and Status Registers that store the various control formats for the device functional definition.

| A <sub>0</sub> | Operation | Register |
|----------------|-----------|----------|
| 0              | Read      | PREG     |
| 0              | Write     | PREG     |
| 1              | Read      | SREG     |
| 1              | Write     | CREG     |

| A <sub>0</sub> | RD | WR | CS |                       |
|----------------|----|----|----|-----------------------|
| 0              | 1  | 0  | 0  | Write 8275H Parameter |
| 0              | 0  | 1  | 0  | Read 8275H Parameter  |
| 1              | 1  | 0  | 0  | Write 8275H Command   |
| 1              | 0  | 1  | 0  | Read 8275H Status     |
| X              | 1  | 1  | 0  | Three-State           |
| X              | X  | X  | 1  | Three-State           |

### $\overline{\text{RD}}$ (READ)

A "low" on this input informs the 8275H that the CPU is reading data or status information from the 8275H.

### $\overline{\text{WR}}$ (WRITE)

A "low" on this input informs the 8275H that the CPU is writing data or control words to the 8275H.

### $\overline{\text{CS}}$ (CHIP SELECT)

A "low" on this input selects the 8275H. No reading or writing will occur unless the device is selected. When  $\overline{\text{CS}}$  is high, the Data Bus is in the float state and RD and WR will have no effect on the chip.

### DRQ (DMA REQUEST)

A "high" on this output informs the DMA Controller that the 8275H desires a DMA transfer.

### $\overline{\text{DACK}}$ (DMA ACKNOWLEDGE)

A "low" on this input informs the 8275H that a DMA cycle is in progress.

### IRQ (INTERRUPT REQUEST)

A "high" on this output informs the CPU that the 8275H desires interrupt service.

### Character Counter

The Character Counter is a programmable counter that is used to determine the number of characters to be displayed per row and the length of the horizontal retrace interval. It is driven by the CCLK (Character Clock) input, which should be a derivative of the external dot clock.

### Line Counter

The Line Counter is a programmable counter that is used to determine the number of horizontal lines (Sweeps) per character row. Its outputs are used to address the external character generator ROM.

### Row Counter

The Row Counter is a programmable counter that is used to determine the number of character rows to be displayed per frame and length of the vertical retrace interval.

### Light Pen Registers

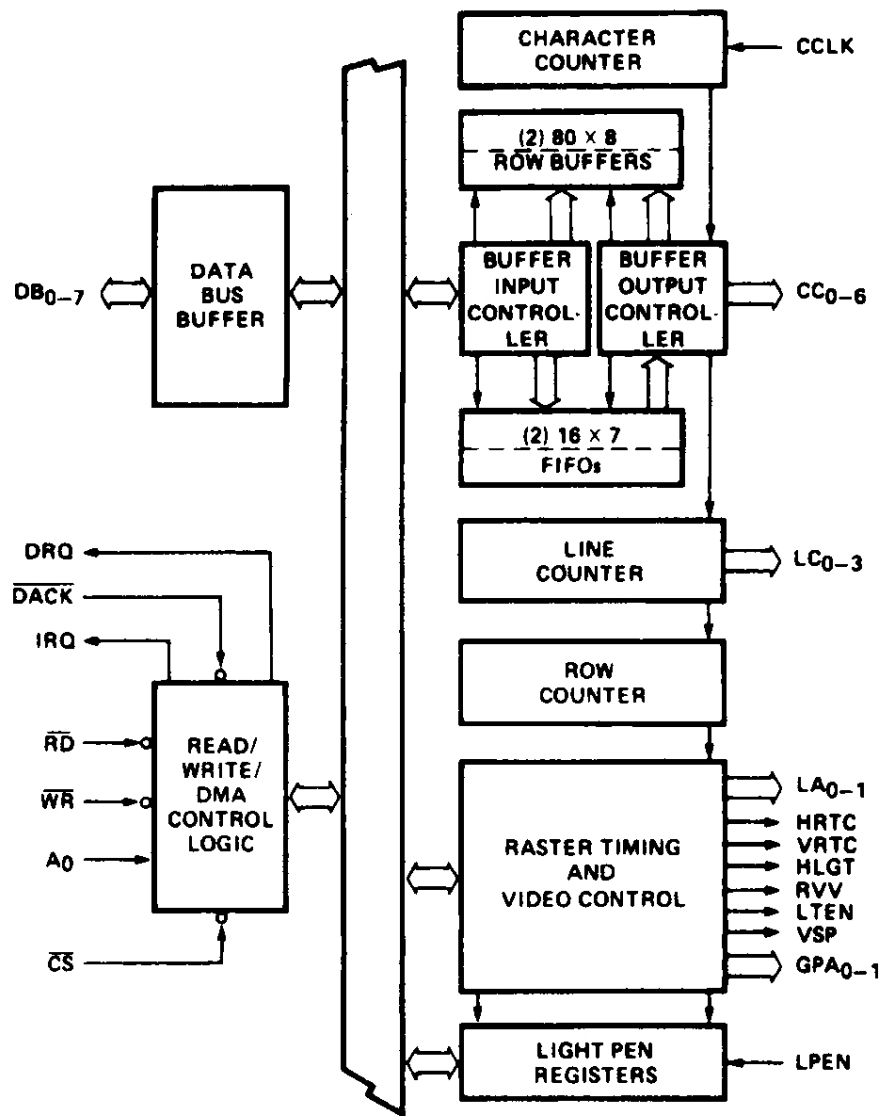
The Light Pen Registers are two registers that store the contents of the character counter and the row counter whenever there is a rising edge on the LPEN (Light Pen) input.

#### NOTE:

Software correction is required.

### Raster Timing and Video Controls

The Raster Timing circuitry controls the timing of the HRTC (Horizontal Retrace) and VRTC (Vertical Retrace) outputs. The Video Control circuitry controls the generation of LA<sub>0-1</sub> (Line Attribute), HGLT (Highlight), RVV (Reverse Video), LTEN (Light Enable), VSP (Video Suppress), and GPA<sub>0-1</sub> (General Purpose Attribute) outputs.



**Figure 3. 8275H Block Diagram Showing Counter and Register Functions**

## Row Buffers

The Row Buffers are two 80 character buffers. They are filled from the microcomputer system memory with the character codes to be displayed. While one row buffer is displaying a row of characters, the other is being filled with the next row of characters.

## FIFOs

There are two 16 character FIFOs in the 8275H. They are used to provide extra row buffer length in the Transparent Attribute Mode (see Detailed Operation section).

## Buffer Input/Output Controllers

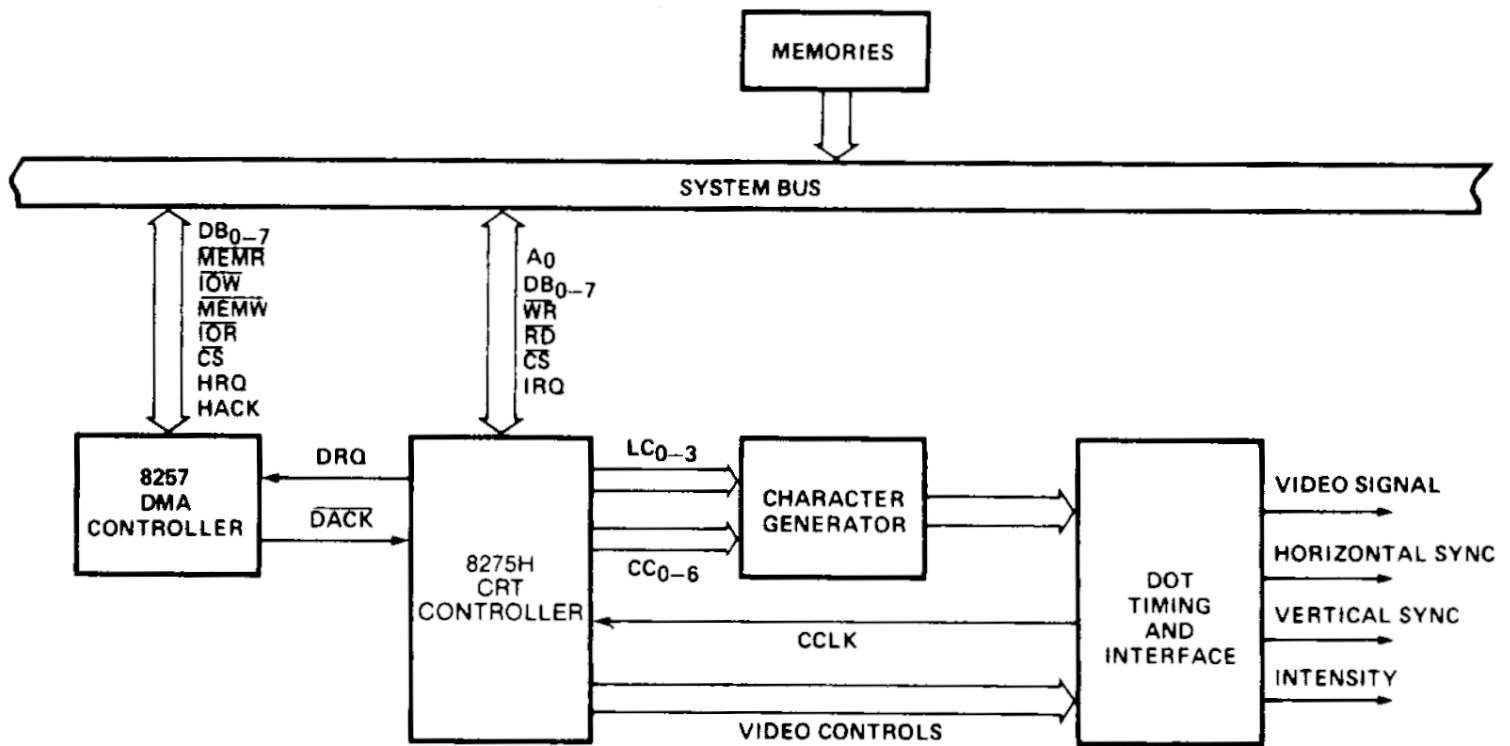
The Buffer Input/Output Controllers decode the characters being placed in the row buffers. If the

character is a character attribute, field attribute or special code, these controllers control the appropriate action. (Examples: An "End of Screen—Stop DMA" special code will cause the Buffer Input Controller to stop further DMA requests. A "Highlight" field attribute will cause the Buffer Output Controller to activate the HGLT output.)

## SYSTEM OPERATION

The 8275H is programmable to a large number of different display formats. It provides raster timing, display row buffering, visual attribute decoding, cursor timing, and light pen detection.

It is designed to interface with the 8257 DMA Controller and standard character generator ROMs for dot matrix decoding. Dot level timing must be provided by external circuitry.



**Figure 4. 8275H Systems Block Diagram Showing Systems Operation**

## General Systems Operational Description

The 8275H provides a "window" into the microcomputer system memory.

Display characters are retrieved from memory and displayed on a row by row basis. The 8275H has two row buffers. While one row buffer is being used for display, the other is being filled with the next row of characters to be displayed. The number of display characters per row and the number of character rows per frame are software programmable, providing easy interface to most CRT displays. (See Programming Section.)

The 8275H requests DMA to fill the row buffer that is not being used for display. DMA burst length and spacing is programmable. (See Programming Section.)

The 8275H displays character rows one line at a time.

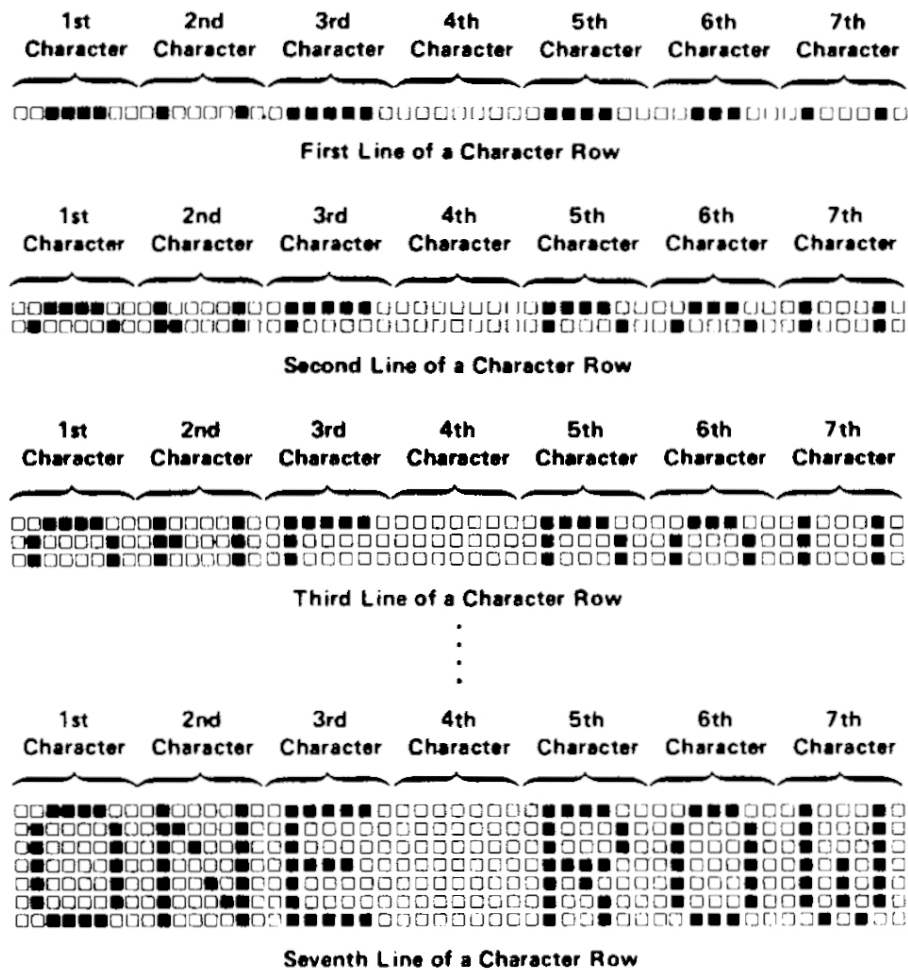
The number of lines per character row, the underline position, and blanking of top and bottom lines are programmable. (See Programming Section.)

The 8275H provides special Control Codes which can be used to minimize DMA or software overhead. It also provides Visual Attribute Codes to cause special action or symbols on the screen without the use of the character generator (See Visual Attributes Section).

The 8275H also controls raster timing. This is done by generating Horizontal Retrace (HRTC) and Vertical Retrace (VRTC) signals. The timing of these signals is programmable.

The 8275H can generate a cursor. Cursor location and format are programmable. (See Programming Section.)

The 8275H has a light pen input and registers. The light pen input is used to load the registers. Light pen registers can be read on command. (See Programming Section.)



**Figure 5. Display of a Character Row**

## Display Row Buffering

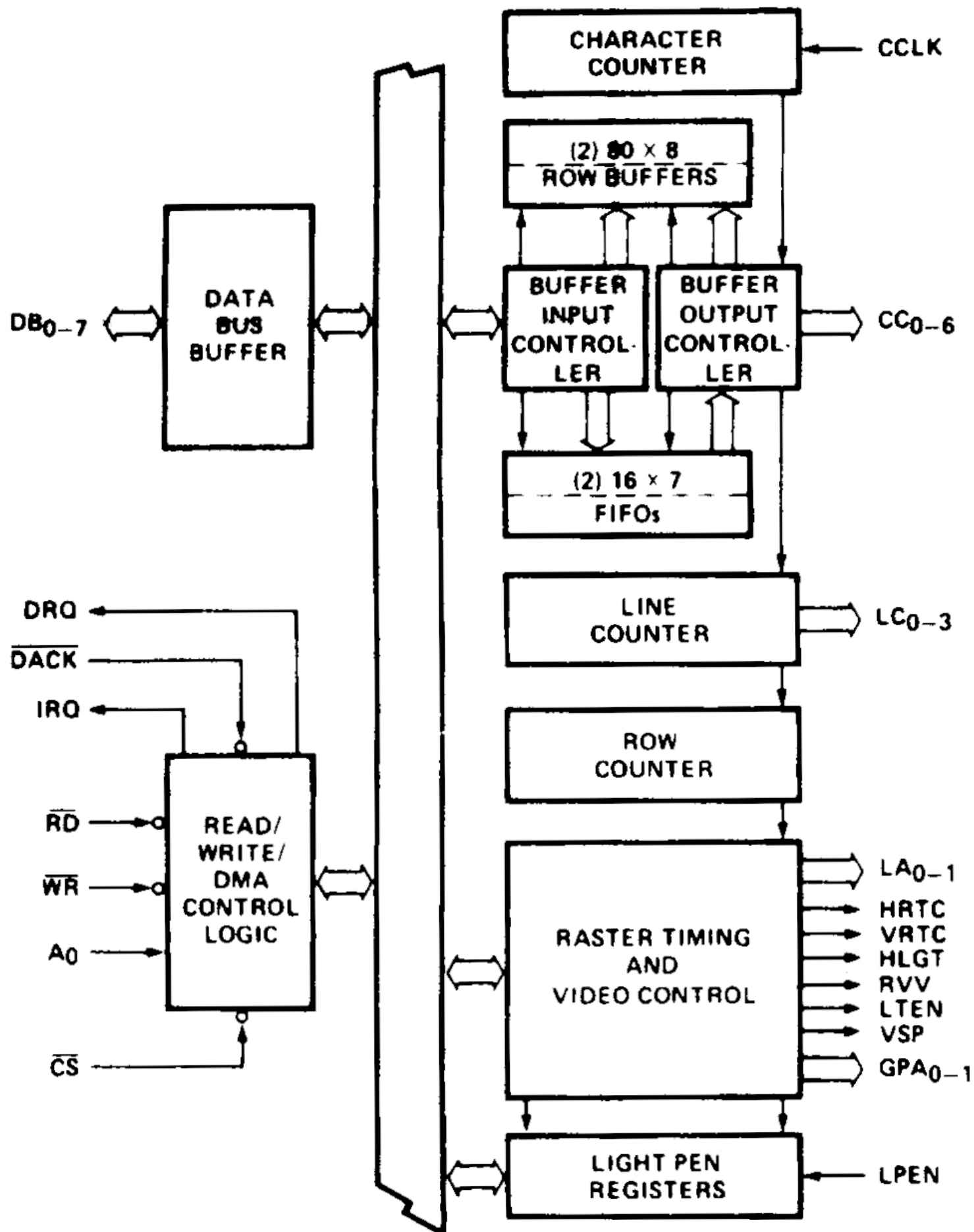
Before the start of a frame, the 8275H requests DMA and one row buffer is filled with characters.

When the first horizontal sweep is started, character codes are output to the character generator from the row buffer just filled. Simultaneously, DMA begins

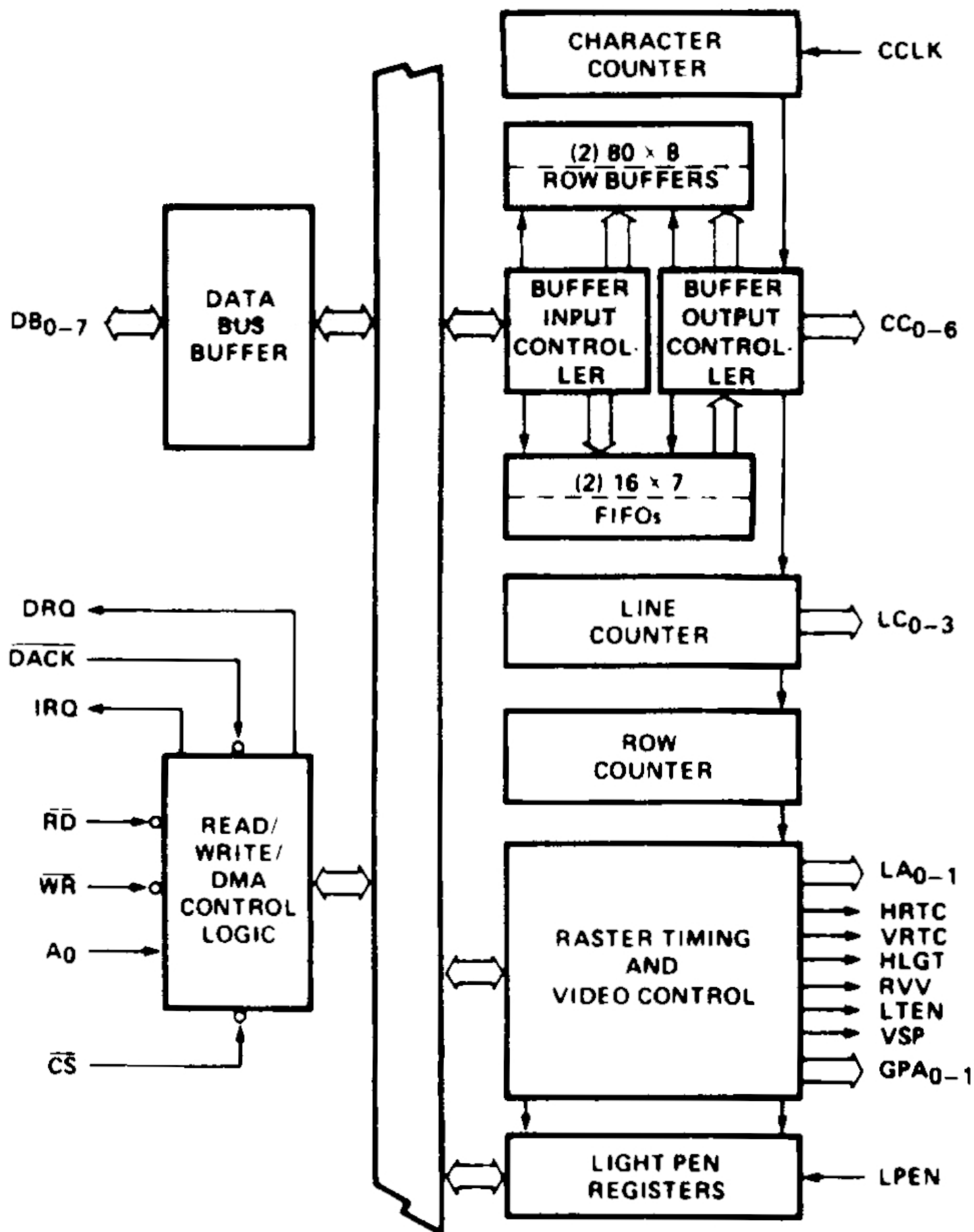
filling the other row buffer with the next row of characters.

After all the lines of the character row are scanned, the roles of the two row buffers are reversed and the same procedure is followed for the next row.

This is repeated until all of the character rows are displayed.



**Figure 6. First Row Buffer Filled**



**Figure 7. Second Buffer Filled, First Row Displayed**

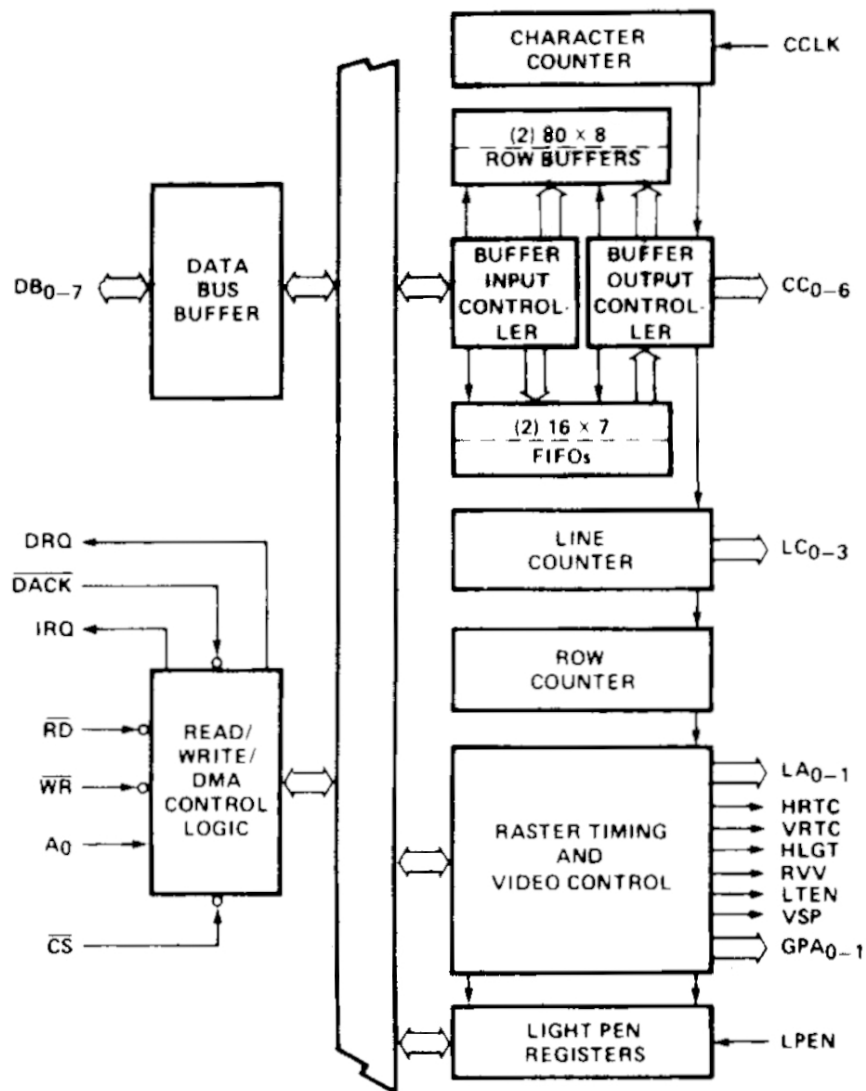


Figure 8. First Buffer Filled with Third Row, Second Row Displayed

## Display Format

### SCREEN FORMAT

The 8275H can be programmed to generate from 1 to 80 characters per row, and from 1 to 64 rows per frame.

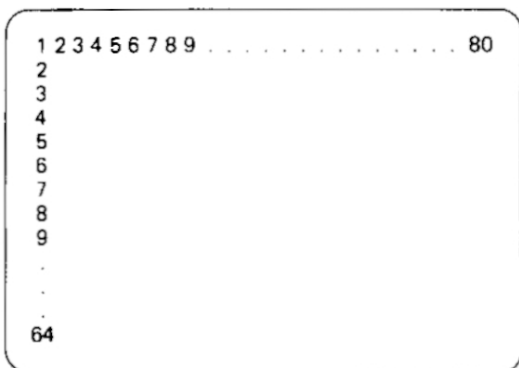


Figure 9. Screen Format

The 8275H can also be programmed to blank alternate rows. In this mode the first row is displayed, the second blanked, the third displayed, etc. DMA is not requested for the blanked rows.

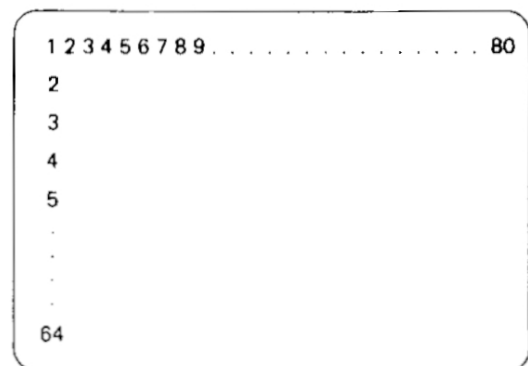


Figure 10. Blank Alternate Rows Mode

## ROW FORMAT

The 8275H is designed to hold the line count stable while outputting the appropriate character codes during each horizontal sweep. The line count is incremented during horizontal retrace and the whole row of character codes are output again during the next sweep. This is continued until the whole character row is displayed.

The number of lines (horizontal sweeps) per character row is programmable from 1 to 16.

The output of the line counter can be programmed to be in one of two modes.

In mode 0, the output of the line *counter* is the same as the line *number*.

In mode 1, the line *counter* is offset by one from the line *number*.

### NOTE:

In mode 1, while the *first* line (line number 0) is being displayed, the *last* count is output by the line counter (see examples).

| Line Number |                 | Line Counter Mode 0 | Line Counter Mode 1 |
|-------------|-----------------|---------------------|---------------------|
| 0           | □ □ □ □ □ □ □ □ | 0000                | 1001                |
| 1           | □ □ □ ■ □ □ □ □ | 0001                | 0000                |
| 2           | □ □ ■ □ □ ■ □ □ | 0010                | 0001                |
| 3           | □ ■ □ □ □ ■ □ □ | 0011                | 0010                |
| 4           | □ ■ □ □ □ ■ □ □ | 0100                | 0011                |
| 5           | □ ■ ■ ■ ■ ■ □ □ | 0101                | 0100                |
| 6           | □ ■ □ □ □ □ ■ □ | 0110                | 0101                |
| 7           | □ ■ □ □ □ □ ■ □ | 0111                | 0110                |
| 8           | □ □ □ □ □ □ □ □ | 1000                | 0111                |
| 9           | □ □ □ □ □ □ □ □ | 1001                | 1000                |

Figure 12. Example of a 10-Line Format

Mode 0 is useful for character generators that leave address zero blank and start at address 1. Mode 1 is useful for character generators which start at address zero.

Underline placement is also programmable (from line number 0 to 15). This is independent of the line counter mode.

If the line *number* of the underline is greater than 7 (line *number* MSB = 1), then the top and bottom lines will be blanked.

| Line Number |                 | Line Counter Mode 0 | Line Counter Mode 1 |
|-------------|-----------------|---------------------|---------------------|
| 0           | □ □ □ □ □ □ □ □ | 0000                | 1111                |
| 1           | □ □ □ □ ■ □ □ □ | 0001                | 0000                |
| 2           | □ □ □ ■ □ ■ □ □ | 0010                | 0001                |
| 3           | □ □ ■ □ □ □ ■ □ | 0011                | 0010                |
| 4           | □ ■ □ □ □ □ ■ □ | 0100                | 0011                |
| 5           | □ ■ □ □ □ □ ■ □ | 0101                | 0100                |
| 6           | □ ■ ■ ■ ■ ■ ■ □ | 0110                | 0101                |
| 7           | □ ■ □ □ □ □ ■ □ | 0111                | 0110                |
| 8           | □ ■ □ □ □ □ ■ □ | 1000                | 0111                |
| 9           | □ ■ □ □ □ □ ■ □ | 1001                | 1000                |
| 10          | □ □ □ □ □ □ □ □ | 1010                | 1001                |
| 11          | □ □ □ □ □ □ □ □ | 1011                | 1010                |
| 12          | □ □ □ □ □ □ □ □ | 1100                | 1011                |
| 13          | □ □ □ □ □ □ □ □ | 1101                | 1100                |
| 14          | □ □ □ □ □ □ □ □ | 1110                | 1101                |
| 15          | □ □ □ □ □ □ □ □ | 1111                | 1110                |

Figure 11. Example of a 16-Line Format

| Line Number |                 | Line Counter Mode 0 | Line Counter Mode 1 |
|-------------|-----------------|---------------------|---------------------|
| 0           | □ □ □ □ □ □ □ □ | 0000                | 1011                |
| 1           | □ □ □ □ ■ □ □ □ | 0001                | 0000                |
| 2           | □ □ □ ■ □ ■ □ □ | 0010                | 0001                |
| 3           | □ □ ■ □ □ □ ■ □ | 0011                | 0010                |
| 4           | □ ■ □ □ □ □ ■ □ | 0100                | 0011                |
| 5           | □ ■ □ □ □ □ ■ □ | 0101                | 0100                |
| 6           | □ ■ ■ ■ ■ ■ ■ □ | 0110                | 0101                |
| 7           | □ ■ □ □ □ □ ■ □ | 0111                | 0110                |
| 8           | □ ■ □ □ □ □ ■ □ | 1000                | 0111                |
| 9           | □ ■ □ □ □ □ ■ □ | 1001                | 1000                |
| 10          | ■ ■ ■ ■ ■ ■ ■ ■ | 1010                | 1001                |
| 11          | □ □ □ □ □ □ □ □ | 1011                | 1010                |

Top and Bottom  
Lines are Blanked

Figure 13. Underline in Line Number 10

If the line *number* of the underline is less than or equal to 7 (line *number* MSB = 0), then the top and bottom lines will *not* be blanked.

| Line Number |               | Line Counter Mode 0 | Line Counter Mode 1 |
|-------------|---------------|---------------------|---------------------|
| 0           | □ □ □ ■ □ □ □ | 0 0 0 0             | 0 1 1 1             |
| 1           | □ □ ■ □ ■ □ □ | 0 0 0 1             | 0 0 0 0             |
| 2           | □ ■ □ □ □ ■ □ | 0 0 1 0             | 0 0 0 1             |
| 3           | □ ■ □ □ □ ■ □ | 0 0 1 1             | 0 0 1 0             |
| 4           | □ ■ ■ □ ■ □ □ | 0 1 0 0             | 0 0 1 1             |
| 5           | □ ■ □ □ □ ■ □ | 0 1 0 1             | 0 1 0 0             |
| 6           | □ ■ □ □ □ ■ □ | 0 1 1 0             | 0 1 0 1             |
| 7           | ■ ■ ■ ■ ■ ■ ■ | 0 1 1 1             | 0 1 1 0             |

Top and Bottom  
Lines are not Blanked

**Figure 14. Underline in Line Number 7**

If the line *number* of the underline is greater than the maximum number of lines, the underline will not appear.

Blanking is accomplished by the VSP (Video Suppression) signal. Underline is accomplished by the LTEN (Light Enable) signal.

## DOT FORMAT

Dot width and character width are dependent upon the external timing and control circuitry.

Dot level timing circuitry should be designed to accept the parallel output of the character generator and shift it out serially at the rate required by the CRT display.

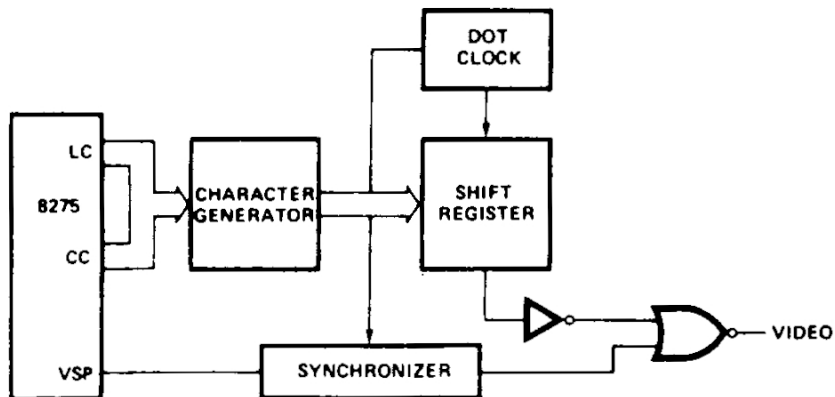
Dot width is a function of dot clock frequency

Character width is a function of the character generator width.

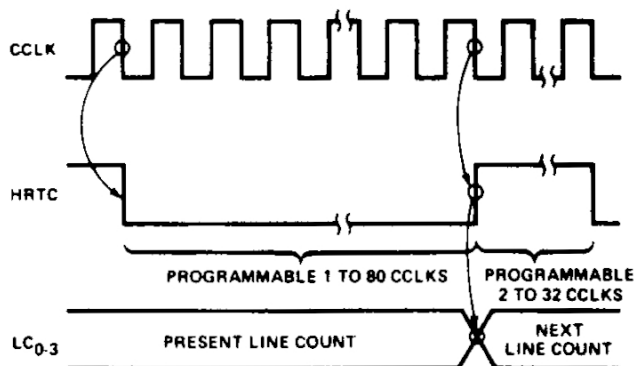
Horizontal character spacing is a function of the shift register length.

## NOTE:

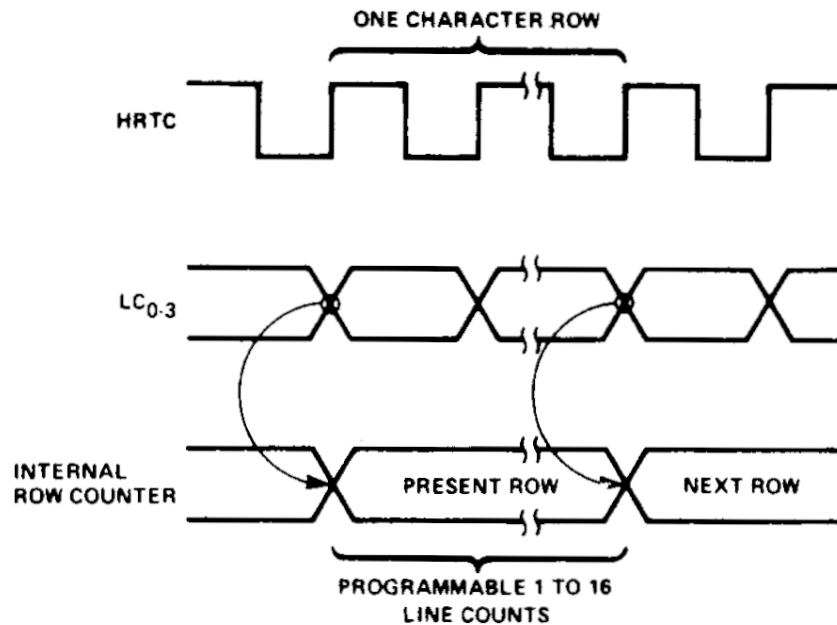
Video control and timing signals must be synchronized with the video signal due to the character generator access delay.



**Figure 15. Typical Dot Level Block Diagram**



**Figure 16. Line Timing**



**Figure 17. Row Timing**

## Raster Timing

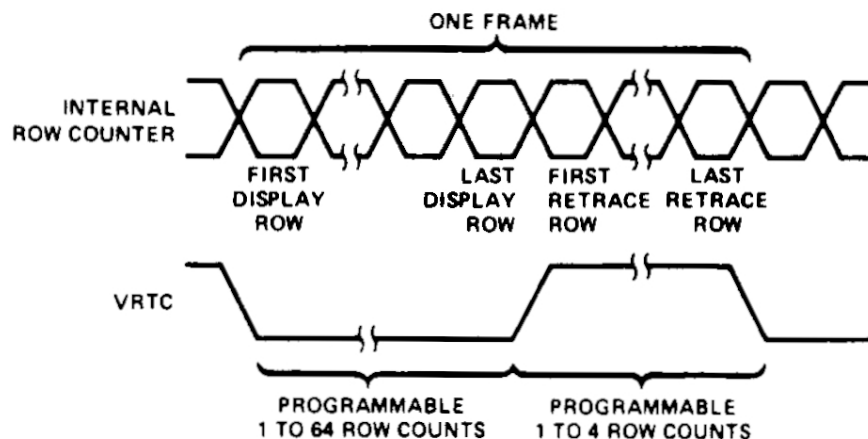
The character counter is driven by the character clock input (CCLK). It counts out the characters being displayed (programmable from 1 to 80). It then causes the line counter to increment, and it starts counting out the horizontal retrace interval (programmable from 2 to 32). This is constantly repeated.

The line counter is driven by the character counter. It is used to generate the line address outputs (LC<sub>0-3</sub>) for the character generator. After it counts all of the lines in a character row (programmable from 1 to 16), it increments the row counter, and starts over

again. (See Character Format Section for detailed description of Line Counter functions.)

The row counter is an internal counter driven by the line counter. It controls the functions of the row buffers and counts the number of character rows displayed.

After the row counter counts all of the rows in a frame (programmable from 1 to 64), it starts counting out the vertical retrace interval (programmable from 1 to 4).



**Figure 18. Frame Timing**

The Video Suppression Output (VSP) is active during horizontal and vertical retrace intervals.

Dot level timing circuitry must synchronize these outputs with the video signal to the CRT Display.

## DMA Timing

The 8275H can be programmed to request burst DMA transfers of 1 to 8 characters. The interval between bursts is also programmable (from 0 to 55 character clock periods  $\pm 1$ ). This allows the user to tailor his DMA overhead to fit his system needs.

The first DMA request of the frame occurs one *row time* before the end of vertical retrace. DMA requests continue as programmed, until the row buffer is filled. If the row buffer is filled in the middle of a burst, the 8275H terminates the burst and resets the burst counter. No more DMA requests will occur until the *beginning* of the *next row*. At that time, DMA requests are activated as programmed until the other buffer is filled.

The first DMA request for a row will start at the first character clock of the preceding row. If the burst

mode is used, the first DMA request may occur a number of character clocks later. This number is equal to the programmed burst space.

If, for any reason, there is a DMA underrun, a flag in the status word will be set.

The DMA controller is typically initialized for the next frame at the end of the current frame.

## Interrupt Timing

The 8275H can be programmed to generate an interrupt request at the end of each frame. This can be used to reinitialize the DMA controller. If the 8275H interrupt enable flag is set, an interrupt request will occur at the *beginning* of the *last display row*.

IRQ will go inactive after the status register is read.

A reset command will also cause IRQ to go inactive, but this is not recommended during normal service.

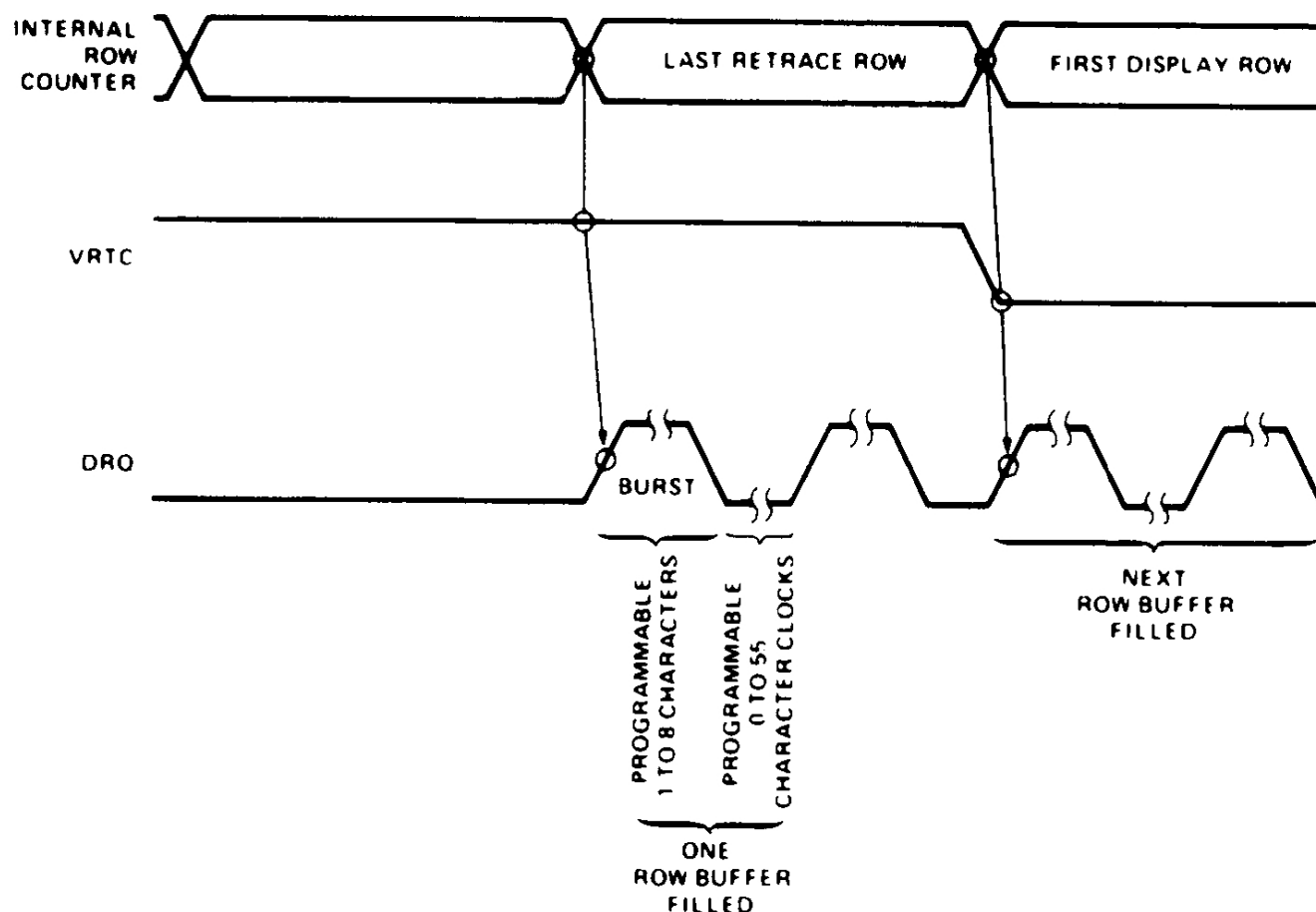
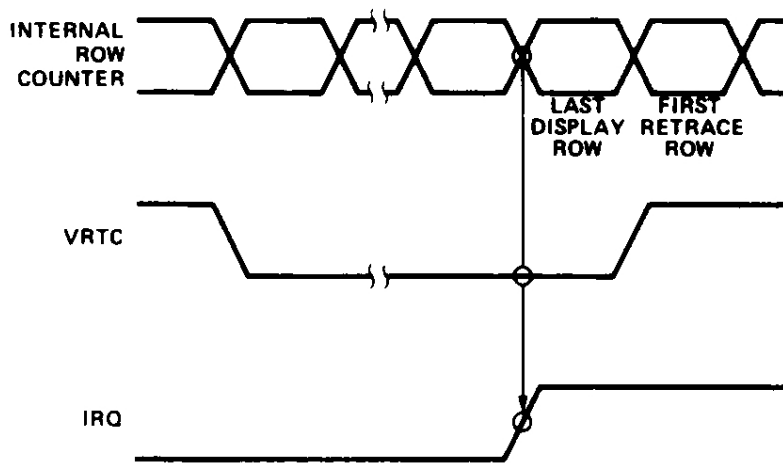
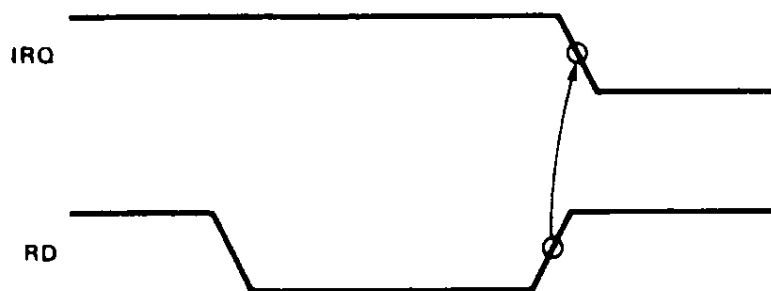


Figure 19. DMA Timing



**Figure 20. Beginning of Interrupt Request**



**Figure 21. End of Interrupt Request**

Another method of reinitializing the DMA controller is to have the DMA controller itself interrupt on terminal count. With this method, the 8275H interrupt enable flag should not be set.

**NOTE:**

Upon power-up, the 8275H Interrupt Enable Flag may be set. As a result, the user's cold start routine should write a reset command to the 8275H before system interrupts are enabled.

## VISUAL ATTRIBUTES AND SPECIAL CODES

The characters processed by the 8275H are 8-bit quantities. The character code outputs provide the character generator with 7 bits of address. The Most Significant Bit is the extra bit and it is used to determine if it is a normal display character (MSB = 0), or if it is a Visual Attribute or Special Code (MSB = 1).

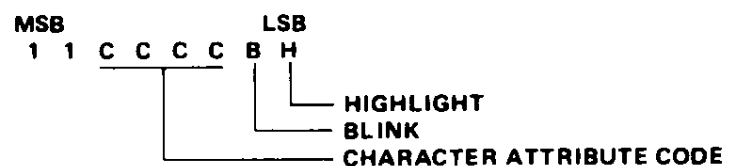
There are two types of Visual Attribute Codes. They are Character Attributes and Field Attributes.

## CHARACTER ATTRIBUTE CODES

Character attribute codes are codes that can be used to generate graphics symbols without the use of a character generator. This is accomplished by selectively activating the Line Attribute outputs ( $LA_0-1$ ), the Video Suppression output (VSP), and the Light Enable output. The dot level timing circuitry can use these signals to generate the proper symbols.

Character attributes can be programmed to blink or be highlighted individually. Blinking is accomplished with the Video Suppression output (VSP). Blink frequency is equal to the screen refresh frequency divided by 32. Highlighting is accomplished by activating the Highlight output (HGLT).

### Character Attributes



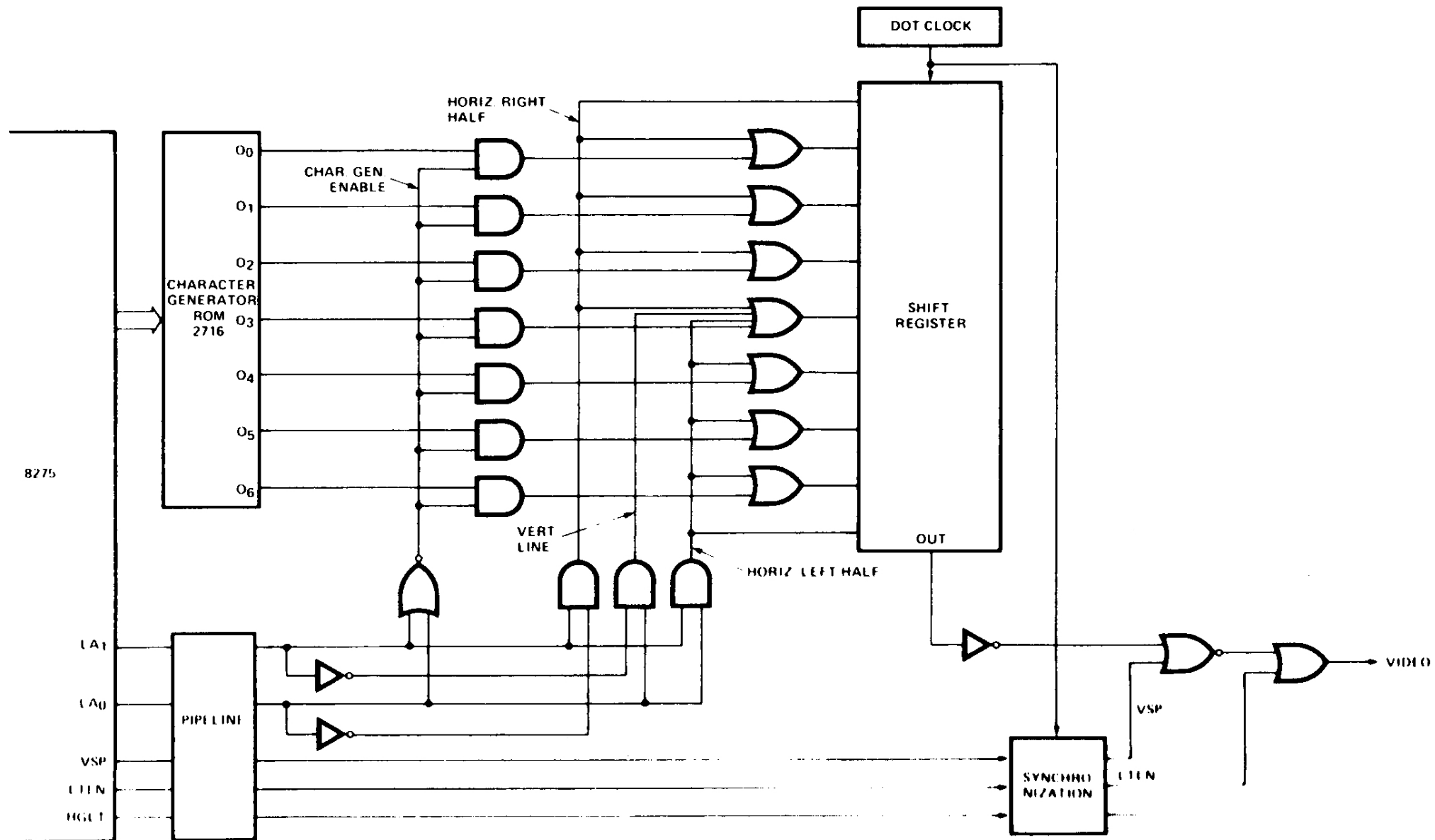













Figure 22. Typical Character Attribute Logic

### Table 2. Character Attributes

Character attributes were designed to produce the following graphics:

| CHARACTER ATTRIBUTE<br>CODE "CCCC" |                 | OUTPUTS         |                 |     |      | SYMBOL  | DESCRIPTION         |
|------------------------------------|-----------------|-----------------|-----------------|-----|------|---|---------------------|
|                                    |                 | LA <sub>1</sub> | LA <sub>0</sub> | VSP | LTEN |   |                     |
| 0000                               | Above Underline | 0               | 0               | 1   | 0    |    | Top Left Corner     |
|                                    | Underline       | 1               | 0               | 0   | 0    |   |                     |
|                                    | Below Underline | 0               | 1               | 0   | 0    |   |                     |
| 0001                               | Above Underline | 0               | 0               | 1   | 0    |    | Top Right Corner    |
|                                    | Underline       | 1               | 1               | 0   | 0    |   |                     |
|                                    | Below Underline | 0               | 1               | 0   | 0    |   |                     |
| 0010                               | Above Underline | 0               | 1               | 0   | 0    |    | Bottom Left Corner  |
|                                    | Underline       | 1               | 0               | 0   | 0    |   |                     |
|                                    | Below Underline | 0               | 0               | 1   | 0    |   |                     |
| 0011                               | Above Underline | 0               | 1               | 0   | 0    |    | Bottom Right Corner |
|                                    | Underline       | 1               | 1               | 0   | 0    |   |                     |
|                                    | Below Underline | 0               | 0               | 1   | 0    |   |                     |
| 0100                               | Above Underline | 0               | 0               | 1   | 0    |    | Top Intersect       |
|                                    | Underline       | 0               | 0               | 0   | 1    |   |                     |
|                                    | Below Underline | 0               | 1               | 0   | 0    |   |                     |
| 0101                               | Above Underline | 0               | 1               | 0   | 0    |    | Right Intersect     |
|                                    | Underline       | 1               | 1               | 0   | 0    |   |                     |
|                                    | Below Underline | 0               | 1               | 0   | 0    |   |                     |
| 0110                               | Above Underline | 0               | 1               | 0   | 0    |    | Left Intersect      |
|                                    | Underline       | 1               | 0               | 0   | 0    |   |                     |
|                                    | Below Underline | 0               | 1               | 0   | 0    |   |                     |
| 0111                               | Above Underline | 0               | 1               | 0   | 0    |    | Bottom Intersect    |
|                                    | Underline       | 0               | 0               | 0   | 1    |   |                     |
|                                    | Below Underline | 0               | 0               | 1   | 0    |   |                     |
| 1000                               | Above Underline | 0               | 0               | 1   | 0    |    | Horizontal Line     |
|                                    | Underline       | 0               | 0               | 0   | 1    |   |                     |
|                                    | Below Underline | 0               | 0               | 1   | 0    |   |                     |
| 1001                               | Above Underline | 0               | 1               | 0   | 0    |  | Vertical Line       |
|                                    | Underline       | 0               | 1               | 0   | 0    |   |                     |
|                                    | Below Underline | 0               | 1               | 0   | 0    |   |                     |
| 1010                               | Above Underline | 0               | 1               | 0   | 0    |  | Crossed Lines       |
|                                    | Underline       | 0               | 0               | 0   | 1    |   |                     |
|                                    | Below Underline | 0               | 1               | 0   | 0    |   |                     |
| 1011                               | Above Underline | 0               | 0               | 0   | 0    |   | Not Recommended *   |
|                                    | Underline       | 0               | 0               | 0   | 0    |   |                     |
|                                    | Below Underline | 0               | 0               | 0   | 0    |   |                     |
| 1100                               | Above Underline | 0               | 0               | 1   | 0    |   | Special Codes       |
|                                    | Underline       | 0               | 0               | 1   | 0    |   |                     |
|                                    | Below Underline | 0               | 0               | 1   | 0    |   |                     |
| 1101                               | Above Underline |                 |                 |     |      |   | Illegal             |
|                                    | Underline       |                 | Undefined       |     |      |   |                     |
|                                    | Below Underline |                 |                 |     |      |   |                     |
| 1110                               | Above Underline |                 |                 |     |      |   | Illegal             |
|                                    | Underline       |                 | Undefined       |     |      |   |                     |
|                                    | Below Underline |                 |                 |     |      |   |                     |
| 1111                               | Above Underline |                 |                 |     |      |   | Illegal             |
|                                    | Underline       |                 | Undefined       |     |      |   |                     |
|                                    | Below Underline |                 |                 |     |      |   |                     |

\*Character Attribute Code 1011 is not recommended for normal operation. Since none of the attribute outputs are active, the character Generator will not be disabled, and an indeterminate character will be generated.

Character Attribute Codes 1101, 1110, and 1111 are illegal.

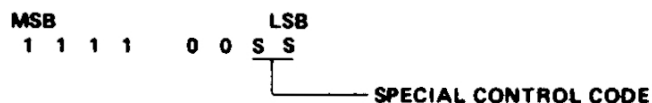
Blinking is active when  $B = 1$ .

Highlight is active when  $H = 1$ .

## SPECIAL CODES

Four special codes are available to help reduce memory, software, or DMA overhead.

### Special Control Character



| S | S | Function               |
|---|---|------------------------|
| 0 | 0 | End of Row             |
| 0 | 1 | End of Row-Stop DMA    |
| 1 | 0 | End of Screen          |
| 1 | 1 | End of Screen-Stop DMA |

The End of Row Code (00) activates VSP and holds it to the end of the line.

The End of Row-Stop DMA Code (01) causes the DMA Control Logic to stop DMA for the rest of the row when it is written into the Row Buffer. It affects the display in the same way as the End of Row Code (00).

The End of Screen Code (10) activates VSP and holds it to the end of the frame.

The End of Screen-Stop DMA Code (11) causes the DMA Control Logic to stop DMA for the rest of the frame when it is written into the Row Buffer. It affects the display in the same way as the End of Screen Code (10).

If the Stop DMA feature is not used, all characters after an End of Row character are ignored, except for the End of Screen character, which operates normally. All characters after an End of Screen character are ignored.

#### NOTE:

If a Stop DMA character is not the last character in a burst or row, DMA is not stopped until after the next character is read. In this situation, a dummy character must be placed in memory after the Stop DMA character.

## FIELD ATTRIBUTES

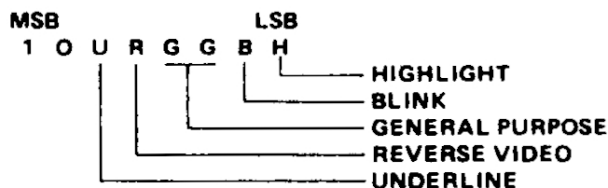
The field attributes are control codes which affect the visual characteristics for a field of characters, starting at the character following the code up to, and including, the character which precedes the *next* field attribute code, or up to the end of the frame. The field attributes are reset during the vertical retrace interval.

There are six field attributes:

- 1) *Blink*—Characters following the code blink by activating the Video Suppression output (VSP). The blink frequency is equal to the screen refresh frequency divided by 32.
- 2) *Highlight*—Characters following the code are highlighted by activating the Highlight output (HGLT).

- 3) *Reverse Video*—Characters following the code appear with reverse video by activating the Reverse Video output (RVV).
- 4) *Underline*—Characters following the code are underlined by activating the Light Enable output (LTEN).
- 5,6) *General Purpose*—There are two additional 8275H outputs which act as general purpose, independently programmable field attributes. GPA<sub>0</sub>–<sub>1</sub> are active high outputs.

## Field Attribute Code



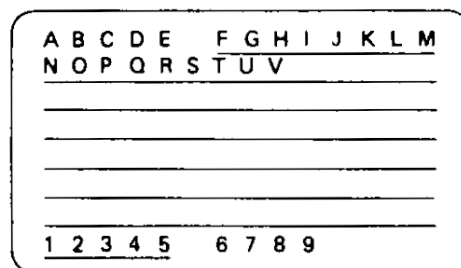
H = 1 for Highlighting  
 B = 1 for Blinking  
 R = 1 for Reverse Video  
 U = 1 for Underline  
 GG = GPA<sub>1</sub>, GPA<sub>0</sub>

#### NOTE:

More than one attribute can be enabled at the same time. If the blinking and reverse video attributes are enabled simultaneously, only the reversed characters will blink.

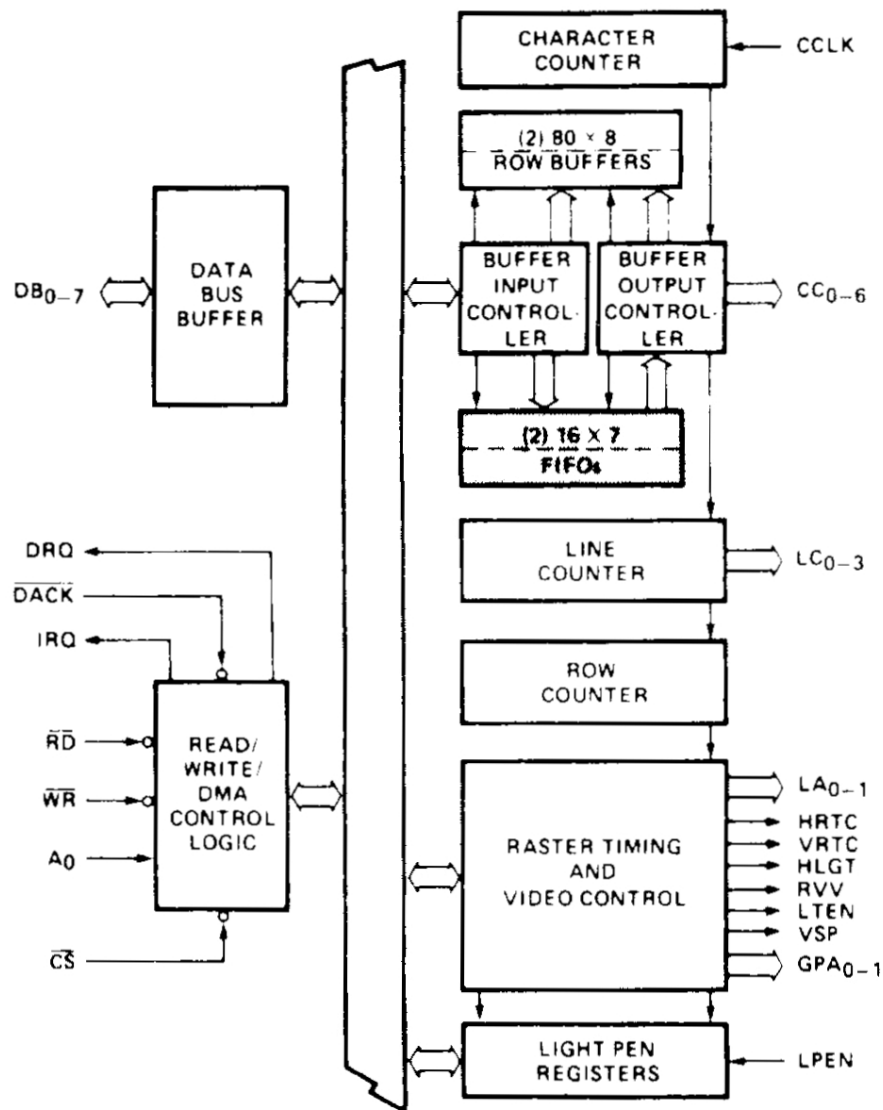
The 8275H can be programmed to provide visible or invisible field attribute characters.

If the 8275H is programmed in the visible field attribute mode, all field attributes will occupy a position on the screen. They will appear as blanks caused by activation of the Video Suppression output (VSP). The chosen visual attributes are activated after this blanked character.



**Figure 23. Example of the Visible Field Attribute Mode (Underline Attribute)**

If the 8275H is programmed in the invisible field attribute mode, the 8275H FIFO is activated.



**Figure 24. Block Diagram Showing FIFO Activation**

Each row buffer has a corresponding FIFO. These FIFOs are 16 characters by 7 bits in size.

When a field attribute is placed in the row buffer during DMA, the buffer input controller recognizes it and places the *next* character in the proper FIFO.

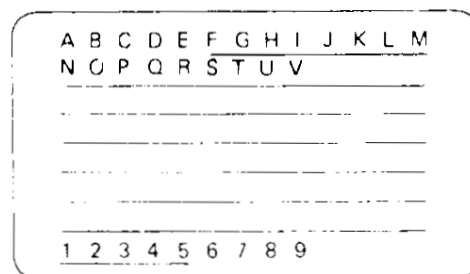
When a field attribute is placed in the Buffer Output Controller during display, it causes the controller to immediately put a character from the FIFO on the Character Code outputs (CC0-6). The chosen Visual Attributes are also activated.

Since the FIFO is 16 characters long, no more than 16 field attribute characters may be used per line in this mode. If more are used, a bit in the status word is set and the first characters in the FIFO are written over and lost.

**NOTE:**

Since the FIFO is 7 bits wide, the MSB of any characters put in it are stripped off. Therefore, a Visual

Attribute or Special Code must *not* immediately follow a field attribute code. If this situation does occur, the Visual Attribute or Special Code will be treated as a normal display character.



**Figure 25. Example of the Invisible Field Attribute Mode (Underline Attribute)**

## Field and Character Attribute Interaction

Character Attribute Symbols are affected by the Reverse Video (RVV) and General Purpose (GPA<sub>0-1</sub>) field attributes. They are not affected by Underline, Blink or Highlight field attributes; however, these characteristics can be programmed *individually* for Character Attribute Symbols.

## Cursor Timing

The cursor location is determined by a cursor row register and a character position register which are loaded by command to the controller. The cursor can be programmed to appear on the display as:

- 1) a blinking underline
- 2) a blinking reverse video block
- 3) a non-blinking underline
- 4) a non-blinking reverse video block

The cursor blinking frequency is equal to the screen refresh frequency divided by 16.

If a non-blinking reverse video *cursor* appears in a non-blinking reverse video *field*, the cursor will appear as a normal video block.

If a non-blinking underline *cursor* appears in a non-blinking underline *field*, the cursor will not be visible.

## Light Pen Detection

A light pen consists of a micro switch and a tiny light sensor. When the light pen is pressed against the CRT screen, the micro switch enables the light sensor. When the raster sweep reaches the light sensor, it triggers the light pen output.

If the output of the light pen is presented to the 8275H LPEN input, the row and character position coordinates are stored in a pair of registers. These registers can be read on command. A bit in the status word is set, indicating that the light pen signal was detected. The LPEN input must be a 0 to 1 transition for proper operation.

## 1.0 Reset Command

|            | Operation | A <sub>0</sub> | Description        | Data Bus |   |   |   |     |   |   |   |
|------------|-----------|----------------|--------------------|----------|---|---|---|-----|---|---|---|
|            |           |                |                    | MSB      |   |   |   | LSB |   |   |   |
| Command    | Write     | 1              | Reset Command      | 0        | 0 | 0 | 0 | 0   | 0 | 0 | 0 |
| Parameters | Write     | 0              | Screen Comp Byte 1 | S        | H | H | H | H   | H | H | H |
|            | Write     | 0              | Screen Comp Byte 2 | V        | V | R | R | R   | R | R | R |
|            | Write     | 0              | Screen Comp Byte 3 | U        | U | U | U | L   | L | L | L |
|            | Write     | 0              | Screen Comp Byte 4 | M        | F | C | C | Z   | Z | Z | Z |

### NOTE:

Due to internal and external delays, the character position coordinate will be off by at least three character positions. This has to be corrected in software.

## Device Programming

The 8275H has two programming registers, the Command Register (CREG) and the parameter register (PREG). It also has a Status Register (SREG). The Command Register can only be written into and the Status Registers can only be read from. They are addressed as follows:

| A <sub>0</sub> | Operation | Register |
|----------------|-----------|----------|
| 0              | Read      | PREG     |
| 0              | Write     | PREG     |
| 1              | Read      | SREG     |
| 1              | Write     | CREG     |

The 8275H expects to receive a command and a sequence of 0 to 4 parameters, depending on the command. If the proper number of parameter bytes are not received before another command is given, a status flag is set, indicating an improper command.

## INSTRUCTION SET

The 8275H instruction set consists of 8 commands.

| Command           | No. of Parameter Bytes |
|-------------------|------------------------|
| Reset             | 4                      |
| Start Display     | 0                      |
| Stop Display      | 0                      |
| Read Light Pen    | 2                      |
| Load Cursor       | 2                      |
| Enable Interrupt  | 0                      |
| Disable Interrupt | 0                      |
| Preset Counters   | 0                      |

In addition, the status of the 8275H (SREG) can be read by the CPU at any time.

**Action**—After the reset command is written, DMA requests stop, 8275H interrupts are disabled, and the VSP output is used to blank the screen. HRTC and VRTC continue to run. HRTC and VRTC timing are random on power-up.

As parameters are written, the screen composition is defined.

**Parameter—S** Spaced Rows

| S | Functions   |
|---|-------------|
| 0 | Normal Rows |
| 1 | Spaced Rows |

**Parameter—HHHHHHH**

Horizontal Characters/Row

| H | H | H | H | H | H | H | No. of Characters Per Row |
|---|---|---|---|---|---|---|---------------------------|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1                         |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2                         |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 3                         |
|   |   |   |   |   |   |   | .                         |
|   |   |   |   |   |   |   | .                         |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 80                        |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | Undefined                 |
|   |   |   |   |   |   |   | .                         |
|   |   |   |   |   |   |   | .                         |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | Undefined                 |

**Parameter—VV** Vertical Retrace Row Count

| V | V | No. of Row Counts Per VRTC |
|---|---|----------------------------|
| 0 | 0 | 1                          |
| 0 | 1 | 2                          |
| 1 | 0 | 3                          |
| 1 | 1 | 4                          |

**Parameter—RRRRRR** Vertical Rows/Frame

| R | R | R | R | R | R | No. of Rows/Frame |
|---|---|---|---|---|---|-------------------|
| 0 | 0 | 0 | 0 | 0 | 0 | 1                 |
| 0 | 0 | 0 | 0 | 0 | 1 | 2                 |
| 0 | 0 | 0 | 0 | 1 | 0 | 3                 |
|   |   |   |   |   |   | .                 |
|   |   |   |   |   |   | .                 |
|   |   |   |   |   |   | .                 |
| 1 | 1 | 1 | 1 | 1 | 1 | 64                |

**Parameter—UUUU** Underline Placement

| U | U | U | U | Line Number of Underline |
|---|---|---|---|--------------------------|
| 0 | 0 | 0 | 0 | 1                        |
| 0 | 0 | 0 | 1 | 2                        |
| 0 | 0 | 1 | 0 | 3                        |
|   |   |   |   | .                        |
|   |   |   |   | .                        |
|   |   |   |   | .                        |
| 1 | 1 | 1 | 1 | 16                       |

**Parameter—LLLL**

Number of Lines per Character Row

| L | L | L | L | No. of Lines/Row |
|---|---|---|---|------------------|
| 0 | 0 | 0 | 0 | 1                |
| 0 | 0 | 0 | 1 | 2                |
| 0 | 0 | 1 | 0 | 3                |
|   |   |   |   | .                |
|   |   |   |   | .                |
|   |   |   |   | .                |
| 1 | 1 | 1 | 1 | 16               |

**Parameter—M** Line Counter Mode

| M | Line Counter Mode          |
|---|----------------------------|
| 0 | Mode 0 (Non-Offset)        |
| 1 | Mode 1 (Offset by 1 Count) |

**Parameter—F** Field Attribute Mode

| F | Field Attribute Mode |
|---|----------------------|
| 0 | Transparent          |
| 1 | Non-Transparent      |

**Parameter—CC Cursor Format**

| C | C | Cursor Format                   |
|---|---|---------------------------------|
| 0 | 0 | Blinking reverse video block    |
| 0 | 1 | Blinking underline              |
| 1 | 0 | Nonblinking reverse video block |
| 1 | 1 | Nonblinking underline           |

**SSS Burst Space Code**

| S | S | S | No. of Character Clocks Between DMA Requests |
|---|---|---|--|
| 0 | 0 | 0 | 0  |
| 0 | 0 | 1 | 7  |
| 0 | 1 | 0 | 15   |
| 0 | 1 | 1 | 23   |
| 1 | 0 | 0 | 31   |
| 1 | 0 | 1 | 39   |
| 1 | 1 | 0 | 47   |
| 1 | 1 | 1 | 55   |

**Parameter—ZZZZ Horizontal Retrace Count**

| Z | Z | Z | Z | No. of Character Counts Per HRTC |
|---|---|---|---|----------------------------------|
| 0 | 0 | 0 | 0 | 2                                |
| 0 | 0 | 0 | 1 | 4                                |
| 0 | 0 | 1 | 0 | 6                                |
|   |   | . |   | .                                |
|   |   | . |   | .                                |
|   |   | . |   | .                                |
| 1 | 1 | 1 | 1 | 32                               |

**BB Burst Count Code**

| B | B | No. of DMA Cycles Per Burst |
|---|---|-----------------------------|
| 0 | 0 | 1                           |
| 0 | 1 | 2                           |
| 1 | 0 | 4                           |
| 1 | 1 | 8                           |

**NOTE:**

uuuu MSB determines blanking of top and bottom lines (1 = blanked, 0 = not blanked).

**Action**—8275H interrupts are enabled, DMA requests begin, video is enabled, Interrupt Enable and Video Enable status flags are set.

**2.0 Start Display Command**

|               | Operation | A <sub>0</sub> | Description   | MSB   | Data Bus | LSB |
|---------------|-----------|----------------|---------------|-------|----------|-----|
| Command       | Write     | 1              | Start Display | 0 0 1 | S S S    | B B |
| No Parameters |           |                |               |       |          |     |

**3.0 Stop Display Command**

|               | Operation | A <sub>0</sub> | Description  | MSB   | Data Bus | LSB   |
|---------------|-----------|----------------|--------------|-------|----------|-------|
| Command       | Write     | 1              | Stop Display | 0 1 0 | 0 0 0    | 0 0 0 |
| No Parameters |           |                |              |       |          |       |

**Action**—Disables video, interrupts remain enabled, HRTC and VRTC continue to run, Video Enable status flag is reset, and the "Start Display" command must be given to re-enable the display.

**4.0 Read Light Pen Command**

|            | Operation | A <sub>0</sub> | Description    | MSB                     | Data Bus | LSB   |
|------------|-----------|----------------|----------------|-------------------------|----------|-------|
| Command    | Write     | 1              | Read Light Pen | 0 1 1                   | 0 0 0    | 0 0 0 |
| Parameters | Read      | 0              | Char. Number   | (Char. Position in Row) |          |       |
|            | Read      | 0              | Row Number     | (Row Number)            |          |       |

**Action**—The 8275H is conditioned to supply the contents of the light pen position registers in the next two read cycles of the parameter register. Status flags are not affected.

**NOTE:**

Software correction of light pen position is required.

## 5.0 Load Cursor Position

|            | Operation      | A <sub>0</sub> | Description                | MSB                                     | Data Bus |   |   |   |   |   | LSB |
|------------|----------------|----------------|----------------------------|---|----------|---|---|---|---|---|-----|
| Command    | Write          | 1              | Load Cursor                | 1                                       | 0        | 0 | 0 | 0 | 0 | 0 | 0   |
| Parameters | Write<br>Write | 0<br>0         | Char. Number<br>Row Number | (Char. Position in Row)<br>(Row Number) |          |   |   |   |   |   |     |

**Action**—The 8275H is conditioned to place the next two parameter bytes into the cursor position registers. Status flags not affected.

## 6.0 Enable Interrupt Command

|               | Operation | A <sub>0</sub> | Description      | MSB | Data Bus |   |   |   |   |   | LSB |
|---------------|-----------|----------------|------------------|-----|----------|---|---|---|---|---|-----|
| Command       | Write     | 1              | Enable Interrupt | 1   | 0        | 1 | 0 | 0 | 0 | 0 | 0   |
| No Parameters |           |                |                  |     |          |   |   |   |   |   |     |

**Action**—The interrupt enable status flag is set and interrupts are enabled.

## 7.0 Disable Interrupt Command

|               | Operation | A <sub>0</sub> | Description       | MSB | Data Bus |   |   |   |   |   | LSB |
|---------------|-----------|----------------|-------------------|-----|----------|---|---|---|---|---|-----|
| Command       | Write     | 1              | Disable Interrupt | 1   | 1        | 0 | 0 | 0 | 0 | 0 | 0   |
| No Parameters |           |                |                   |     |          |   |   |   |   |   |     |

**Action**—Interrupts are disabled and the interrupt enable status flag is reset.

## 8.0 Preset Counters Command

|               | Operation | A <sub>0</sub> | Description     | MSB | Data Bus |   |   |   |   |   | LSB |
|---------------|-----------|----------------|-----------------|-----|----------|---|---|---|---|---|-----|
| Command       | Write     | 1              | Preset Counters | 1   | 1        | 1 | 0 | 0 | 0 | 0 | 0   |
| No Parameters |           |                |                 |     |          |   |   |   |   |   |     |

**Action**—The internal timing counters are preset, corresponding to a screen display position at the top left corner. Two character clocks are required for this operation. The counters will remain in this state until any other command is given.

This command is useful for system debug and synchronization of clustered CRT displays on a single CPU. After this command, two additional clock cycles are required before the first character of the first row is put out.

## STATUS FLAGS

|         | Operation | A <sub>0</sub> | Description | Data Bus |    |    |    |     |    |    |    |
|---------|-----------|----------------|-------------|----------|----|----|----|-----|----|----|----|
|         |           |                |             | MSB      |    |    |    | LSB |    |    |    |
| Command | Read      | 1              | Status Word | 0        | IE | IR | LP | IC  | VE | DU | F0 |

IE — (Interrupt Enable) Set or reset by command. It enables vertical retrace interrupt. It is automatically set by a "Start Display" command and reset with the "Reset" command.

IR — (Interrupt Request) This flag is set at the beginning of display of the last row of the frame if the interrupt enable flag is set. It is reset after a status read operation.

LP — This flag is set when the light pen input (LPEN) is activated and the light pen registers have been loaded. This flag is automatically reset after a status read.

IC — (Improper Command) This flag is set when a command parameter string is too long or

too short. The flag is automatically reset after a status read.

VE — (Video Enable) This flag indicates that video operation of the CRT is enabled. This flag is set on a "Start Display" command, and reset on a "Stop Display" or "Reset" command.

DU — (DMA Underrun) This flag is set whenever a data underrun occurs during DMA transfers. Upon detection of DU, the DMA operation is stopped and the screen is blanked until after the vertical retrace interval. This flag is reset after a status read.

FO — (FIFO Overrun) This flag is set whenever the FIFO is overrun. It is reset on a status read.

## ABSOLUTE MAXIMUM RATINGS\*

Ambient Temperature Under Bias . . . . . 0°C to 70°C

Storage Temperature . . . . . -65°C to +150°C

Voltage On Any Pin

With Respect to Ground . . . . . -0.5V to +7V

Power Dissipation . . . . . 1 Watt

*\*Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

## D.C. CHARACTERISTICS $T_A = 0^\circ\text{C to } 70^\circ\text{C}, V_{CC} = 5\text{V} \pm 5\%$

| Symbol           | Parameter                      | Min  | Max                    | Units   | Test Conditions                             |
|------------------|--------------------------------|------|------------------------|---------|---|
| V <sub>IL</sub>  | Input Low Voltage              | -0.5 | 0.8                    | V       |   |
| V <sub>IH</sub>  | Input High Voltage             | 2.0  | V <sub>CC</sub> + 0.5V | V       |   |
| V <sub>OL</sub>  | Output Low Voltage             |      | 0.45                   | V       | I <sub>OL</sub> = 2.2 mA                    |
| V <sub>OH</sub>  | Output High Voltage            | 2.4  |                        | V       | I <sub>OH</sub> = -400 $\mu$ A              |
| I <sub>IL</sub>  | Input Load Current             |      | $\pm 10$               | $\mu$ A | V <sub>IN</sub> = V <sub>CC</sub> to 0V     |
| I <sub>OFL</sub> | Output Float Leakage           |      | $\pm 10$               | $\mu$ A | V <sub>OUT</sub> = V <sub>CC</sub> to 0.45V |
| I <sub>CC</sub>  | V <sub>CC</sub> Supply Current |      | 160                    | mA      |   |

**CAPACITANCE**  $T_A = 25^{\circ}\text{C}$ ,  $V_{CC} = \text{GND} = 0\text{V}$ 

| Symbol    | Parameter         | Min | Max | Units | Test Conditions  |
|-----------|-------------------|-----|-----|-------|--|
| $C_{IN}$  | Input Capacitance |     | 10  | pF    | $f_c = 1\text{ MHz}$<br>Unmeasured pins returned to $V_{SS}$ |
| $C_{I/O}$ | I/O Capacitance   |     | 20  | pF    |  |

**A.C. CHARACTERISTICS**  $T_A = 0^{\circ}\text{C}$  to  $70^{\circ}\text{C}$ ,  $V_{CC} = 5.0\text{V} \pm 5\%$ ,  $\text{GND} = 0\text{V}$ **Bus Parameters****READ CYCLE**

| Symbol   | Parameter                  | Min | Max | Units | Test Conditions       |
|----------|----------------------------|-----|-----|-------|-----------------------|
| $t_{AR}$ | Address Stable before READ | 0   |     | ns    |                       |
| $t_{RA}$ | Address Hold Time for READ | 0   |     | ns    |                       |
| $t_{RR}$ | READ Pulse Width           | 250 |     | ns    |                       |
| $t_{RD}$ | Data Delay from READ       |     | 200 | ns    | $C_L = 150\text{ pF}$ |
| $t_{DF}$ | READ to Data Floating      |     | 100 | ns    | $C_L = 150\text{ pF}$ |

**WRITE CYCLE**

| Symbol   | Parameter                   | Min | Max | Units | Test Conditions |
|----------|-----------------------------|-----|-----|-------|-----------------|
| $t_{AW}$ | Address Stable before WRITE | 0   |     | ns    |                 |
| $t_{WA}$ | Address Hold Time for WRITE | 0   |     | ns    |                 |
| $t_{WW}$ | WRITE Pulse Width           | 250 |     | ns    |                 |
| $t_{DW}$ | Data Setup Time for WRITE   | 150 |     | ns    |                 |
| $t_{WD}$ | Data Hold Time for WRITE    | 0   |     | ns    |                 |

**CLOCK TIMING**

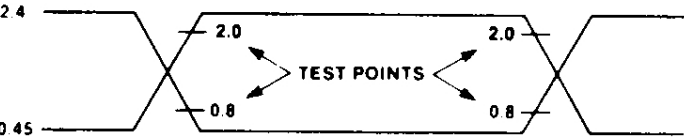
| Symbol    | Parameter    | 8275 |     | 8275-2 |     | Units | Test Conditions |
|-----------|--------------|------|-----|--------|-----|-------|-----------------|
|           |              | Min  | Max | Min    | Max |       |                 |
| $t_{CLK}$ | Clock Period | 480  |     | 320    |     | ns    |                 |
| $t_{KH}$  | Clock High   | 240  |     | 120    |     | ns    |                 |
| $t_{KL}$  | Clock Low    | 160  |     | 120    |     | ns    |                 |
| $t_{KR}$  | Clock Rise   | 5    | 30  | 5      | 30  | ns    |                 |
| $t_{KF}$  | Clock Fall   | 5    | 30  | 5      | 30  | ns    |                 |

OTHER TIMING

| Symbol          | Parameter                       | 8275 |     | 8275-2 |     | Units | Test Conditions        |
|-----------------|---------------------------------|------|-----|--------|-----|-------|------------------------|
|                 |                                 | Min  | Max | Min    | Max |       |                        |
| t <sub>CC</sub> | Character Code Output Delay     |      | 150 |        | 150 | ns    | C <sub>L</sub> = 50 pF |
| t <sub>HR</sub> | Horizontal Retrace Output Delay |      | 200 |        | 150 | ns    | C <sub>L</sub> = 50 pF |
| t <sub>LC</sub> | Line Count Output Delay         |      | 400 |        | 250 | ns    | C <sub>L</sub> = 50 pF |
| t <sub>AT</sub> | Control/Attribute Output Delay  |      | 275 |        | 250 | ns    | C <sub>L</sub> = 50 pF |
| t <sub>VR</sub> | Vertical Retrace Output Delay   |      | 275 |        | 250 | ns    | C <sub>L</sub> = 50 pF |
| t <sub>RI</sub> | IRQ ↓ from RD ↑                 |      | 250 |        | 250 | ns    | C <sub>L</sub> = 50 pF |
| t <sub>WQ</sub> | DRQ ↑ from WR ↑                 |      | 250 |        | 250 | ns    | C <sub>L</sub> = 50 pF |
| t <sub>RQ</sub> | DRQ ↓ from WR ↓                 |      | 200 |        | 200 | ns    | C <sub>L</sub> = 50 pF |
| t <sub>LR</sub> | DACK ↓ to WR ↓                  | 0    |     | 0      |     | ns    |                        |
| t <sub>RL</sub> | WR ↑ to DACK ↑                  | 0    |     | 0      |     | ns    |                        |
| t <sub>PR</sub> | LPEN Rise                       |      | 50  |        | 50  | ns    |                        |
| t <sub>PH</sub> | LPEN Hold                       | 100  |     | 100    |     | ns    |                        |
| t <sub>DI</sub> | DACK Inactive Period            | 120  |     |        |     | ns    |                        |

A.C. TESTING INPUT, OUTPUT WAVEFORM

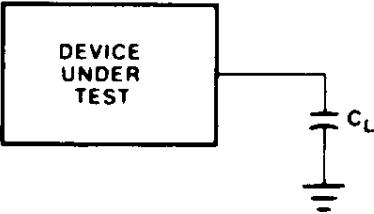
INPUT/OUTPUT



210464-30

A.C. Testing: Inputs are driven at 2.4V for a Logic "1" and 0.45V for a Logic "0." Timing measurements are made at 2.0V for a Logic "1" and 0.8V for a Logic "0."

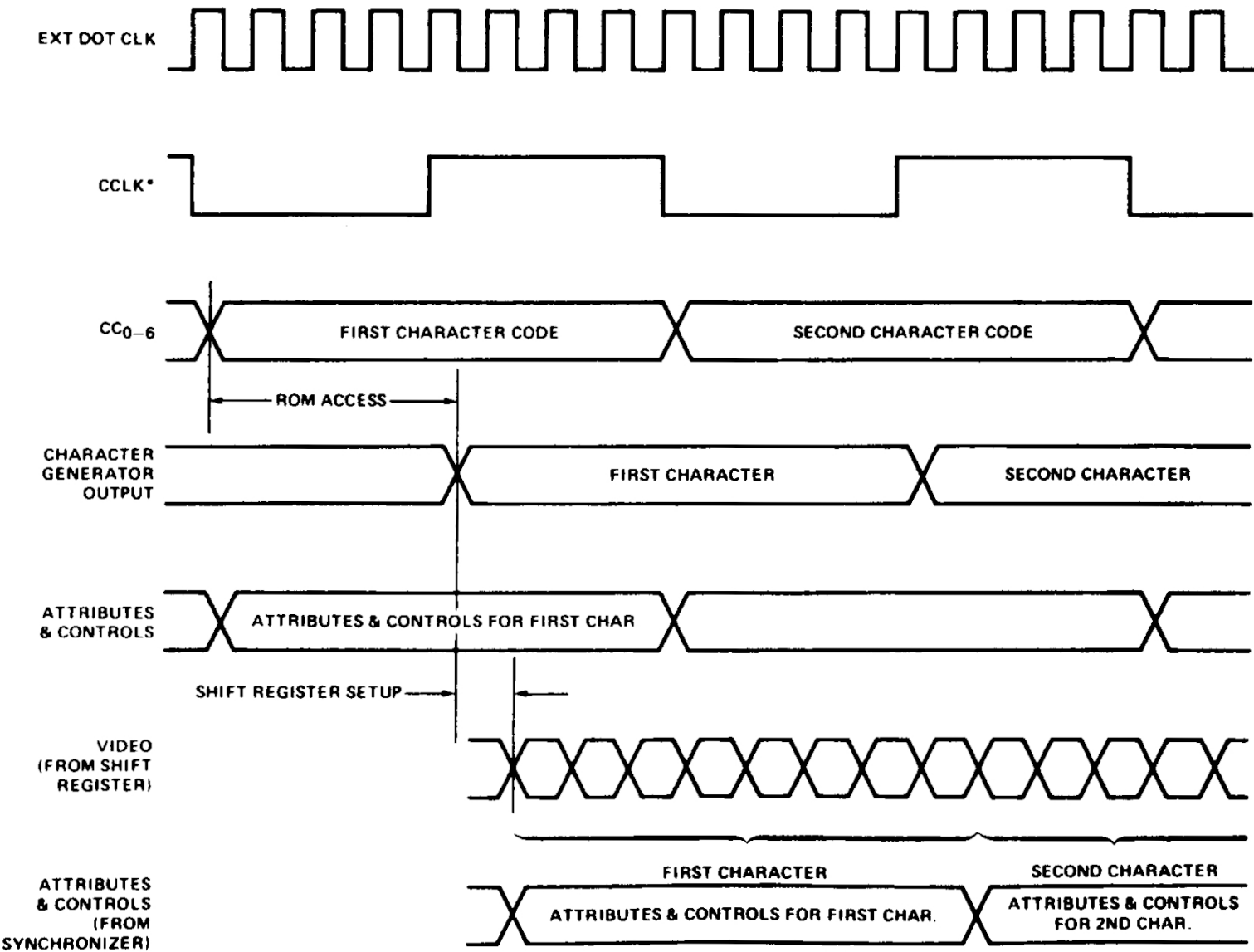
A.C. TESTING LOAD CIRCUIT



C<sub>L</sub> Includes Jig Capacitance

# WAVEFORMS

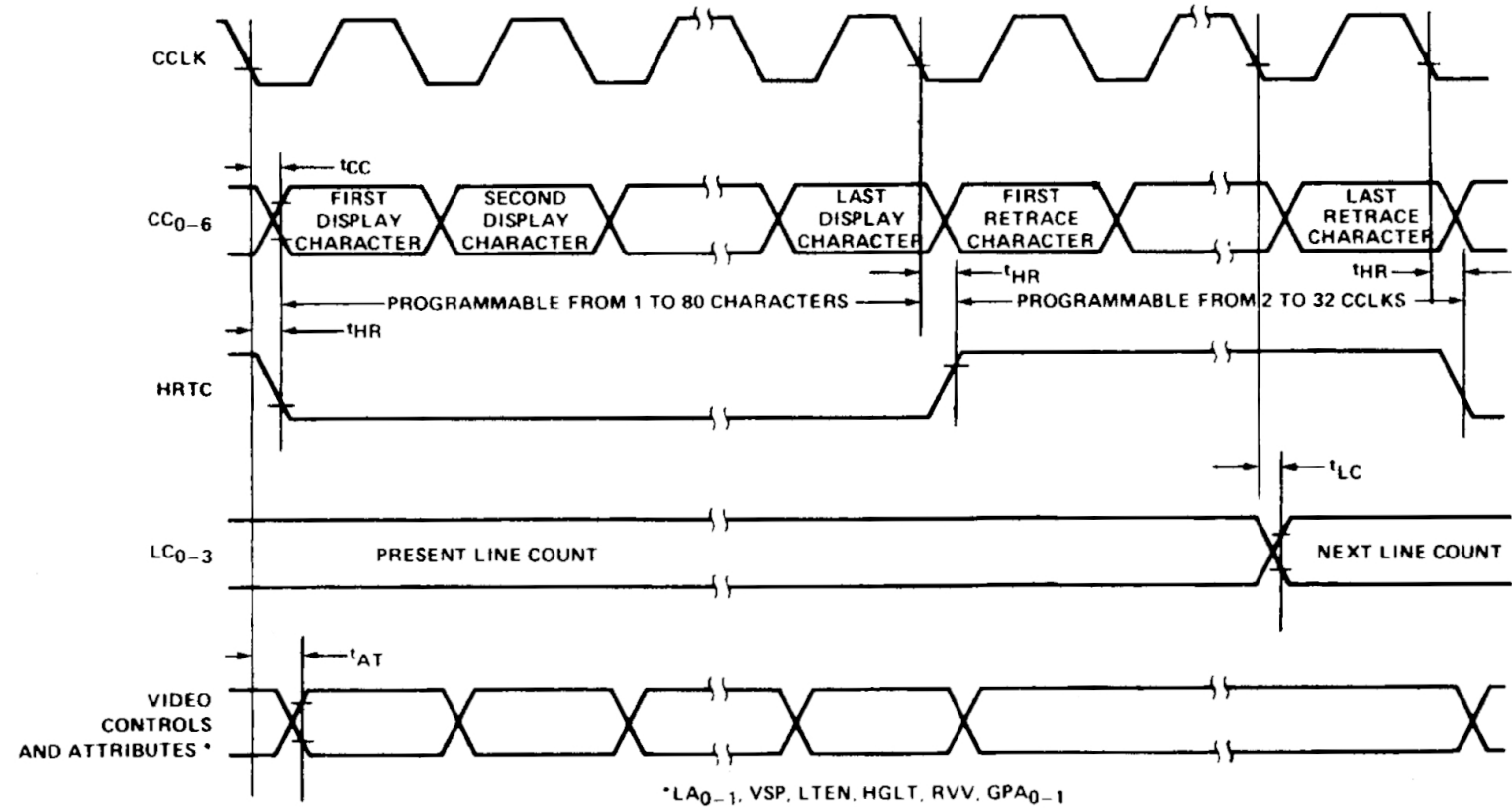
## TYPICAL DOT LEVEL TIMING



\*CCLK IS A MULTIPLE OF THE DOT CLOCK AND AN INPUT TO THE 8275.

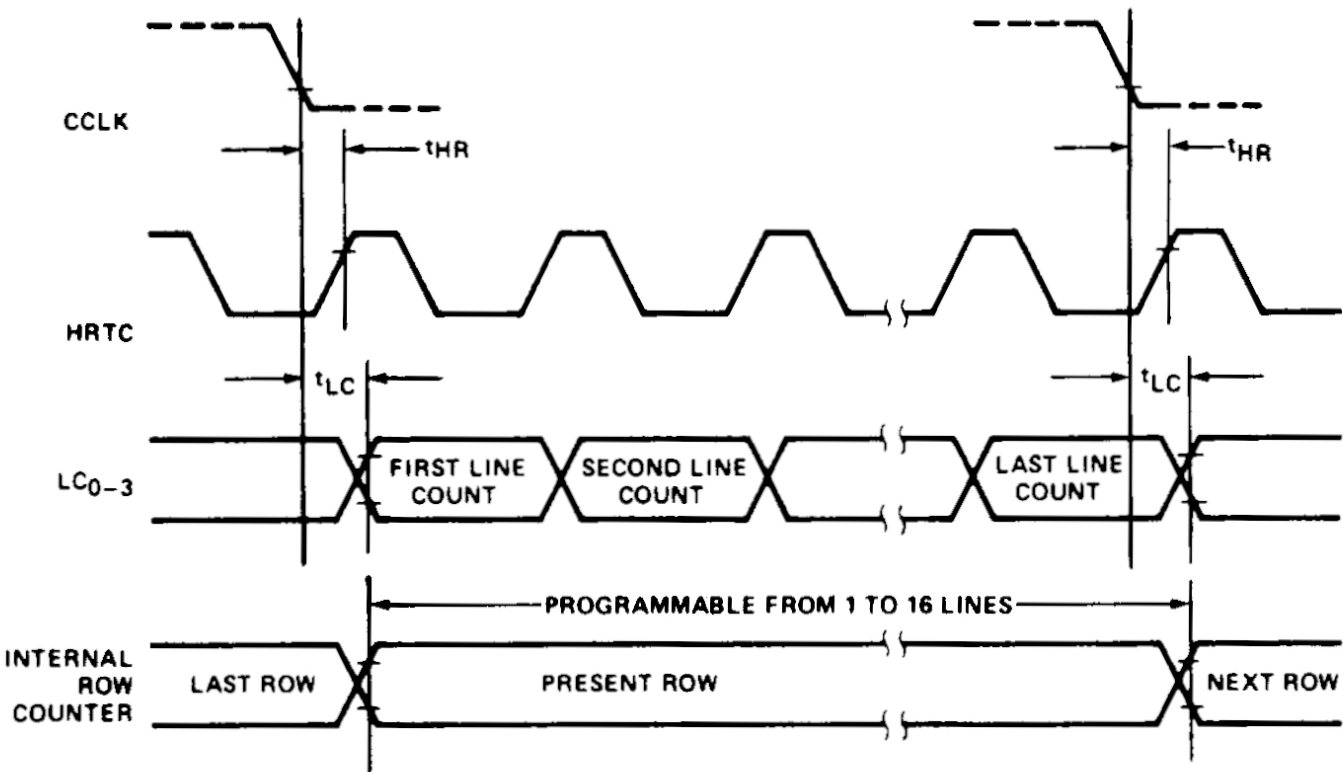
WAVEFORMS (Continued)

LINE TIMING

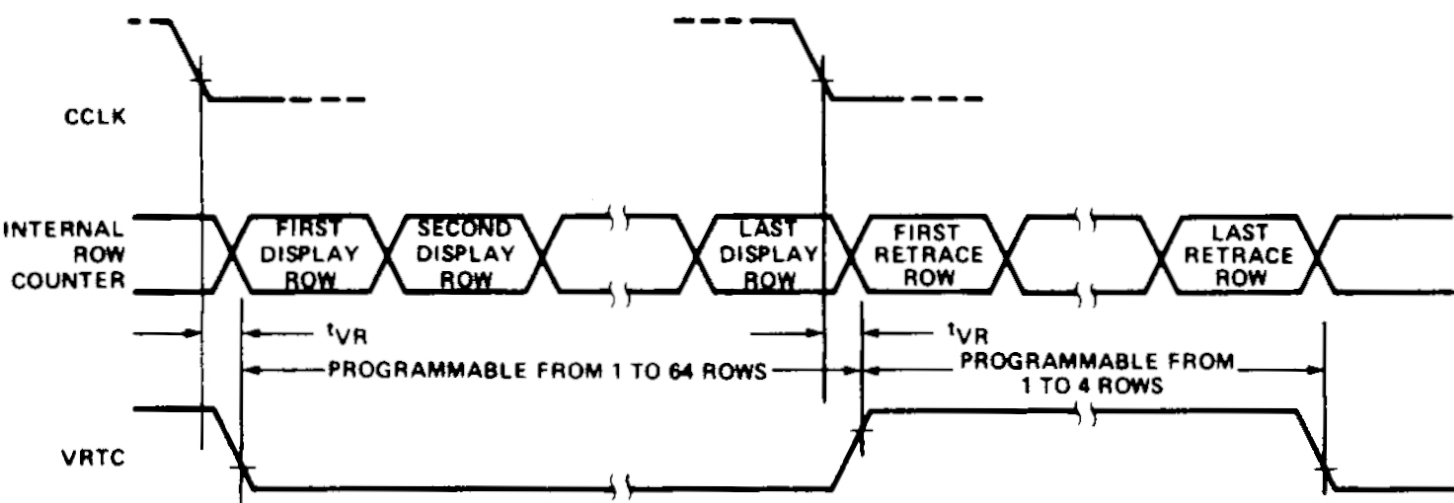


WAVEFORMS (Continued)

ROW TIMING

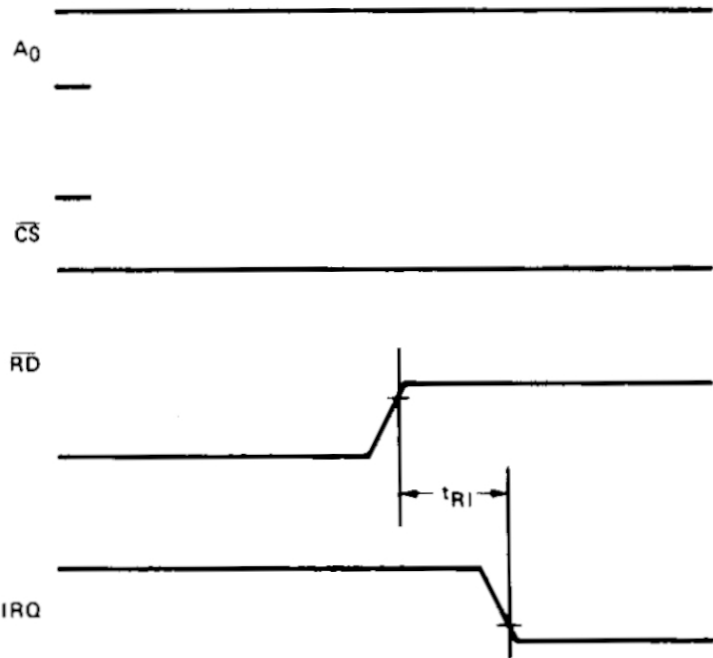


FRAME TIMING

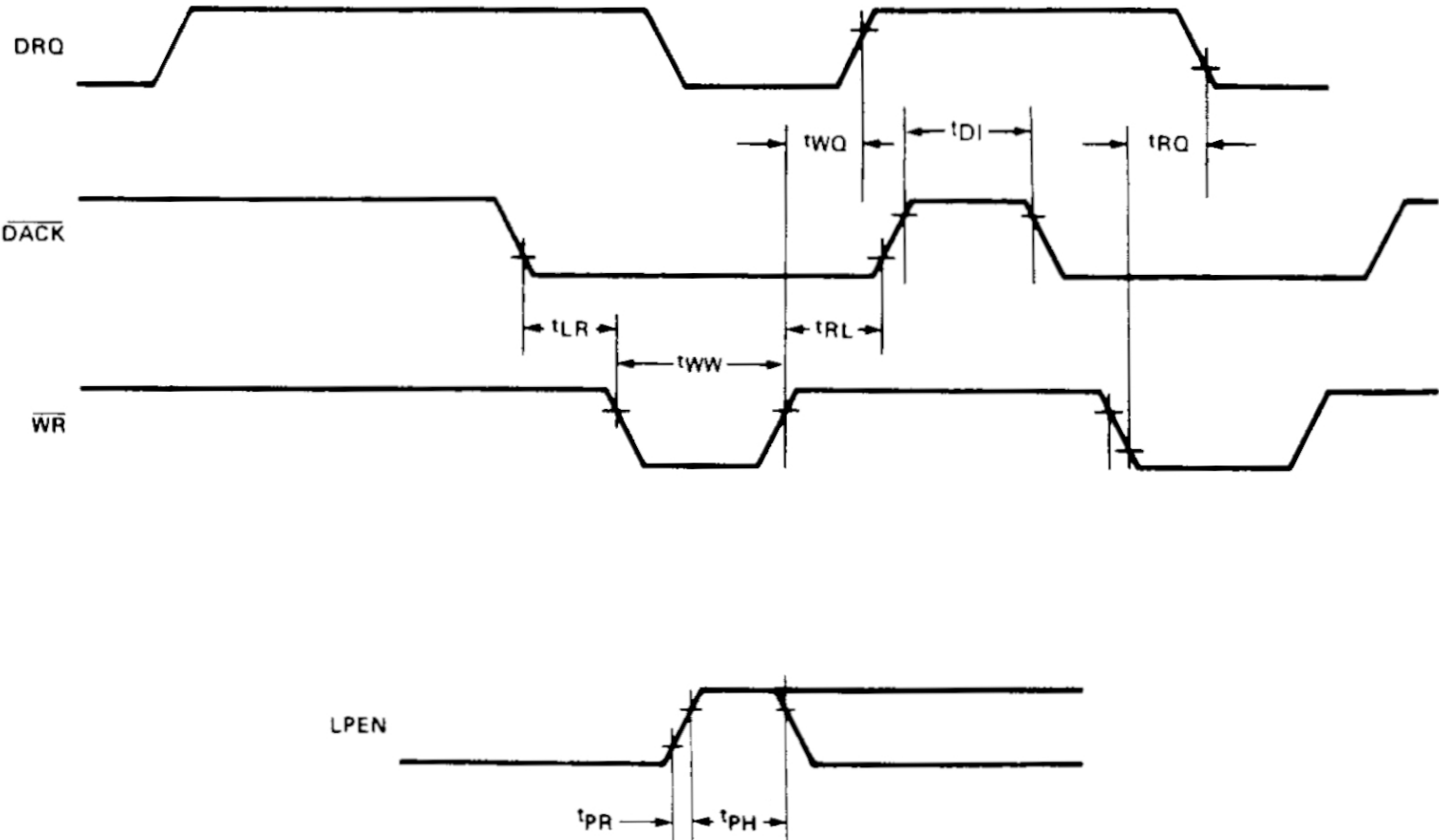


WAVEFORMS (Continued)

INTERRUPT TIMING

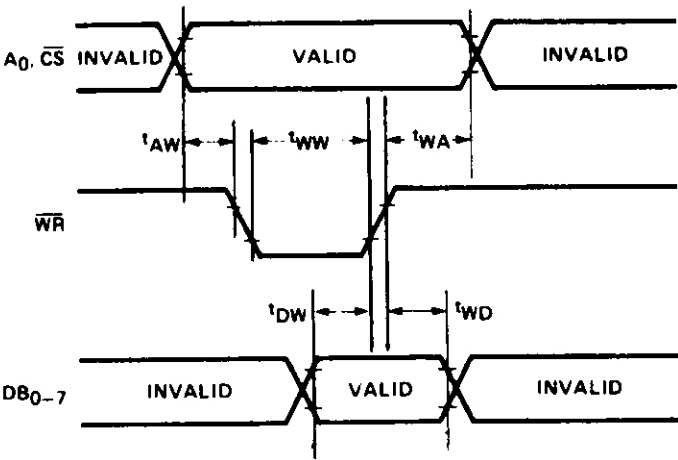


DMA TIMING

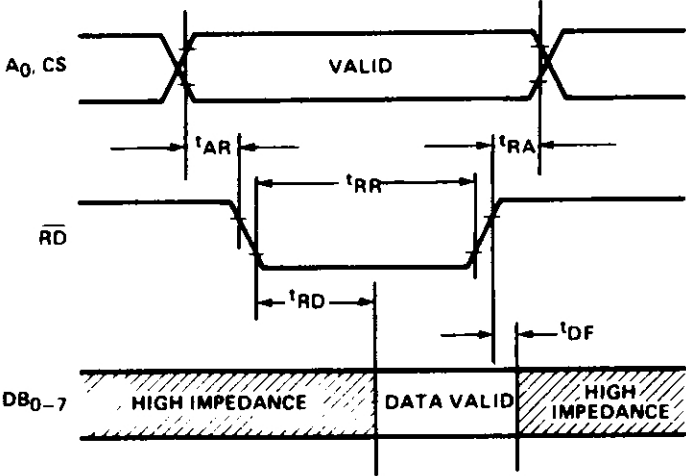


# WAVEFORMS (Continued)

## WRITE TIMING



## READ TIMING



## CLOCK TIMING

