STANDARD MICROSYSTEMS CORPORATION

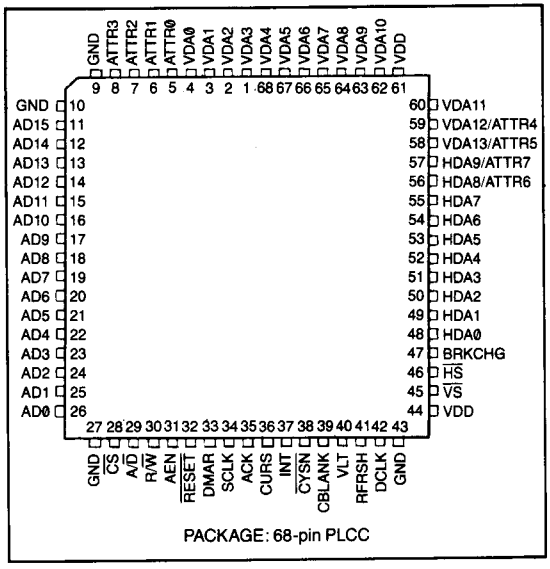
CRT97C11
PRELIMINARY

Video Engine For Windows VIEW

FEATURES:

- ☐ 127 Independent Windows Max per Screen.
- ☐ Windows Specified Relative to Screen:
  - Window Number
  - X-Y Start/End Screen Coordinate
  - X-Y Display Memory Start Address
- ☐ Attributes can be Specified on a per Window Basis:
  - Window Priority
  - Up to 8 Parallel Attributes/Window Output by VIEW
  - Background windows
- ☐ Three Internal Break Address Buffers.
- ☐ 32 Max Visible Horizontal Breaks per Scan Line.
- ☐ Capable of Generating Screen Resolutions as High as 4K x 4K Pixels (assuming a 16 bit wide display memory).
- ☐ Private Display and System Buses:
  - 20 Bit X-Y Display Memory Address Bus + 4 Bit Extended Bus
  - 16 Bit Address/Data Bus to System Memory
- ☐ Separate Clocks for Display and System Buses.
- ☐ DMA Master Capability for Interfacing to System Memory.
- ☐ Automatic and Transparent Dynamic RAM Refresh for Display Memory.
- ☐ Programmable Cursor Output.
- ☐ Fully Programmable Display Format.
- ☐ Normal or Interlaced Video Output.

PIN CONFIGURATION



PACKAGE: 68-pin PLCC

- ☐ Horizontal and Vertical Drive Signals may be Externally Synchronized.
- ☐ Compatible with 680X0 and 80X86 Processors.
- ☐ Low Power CMOS Technology.

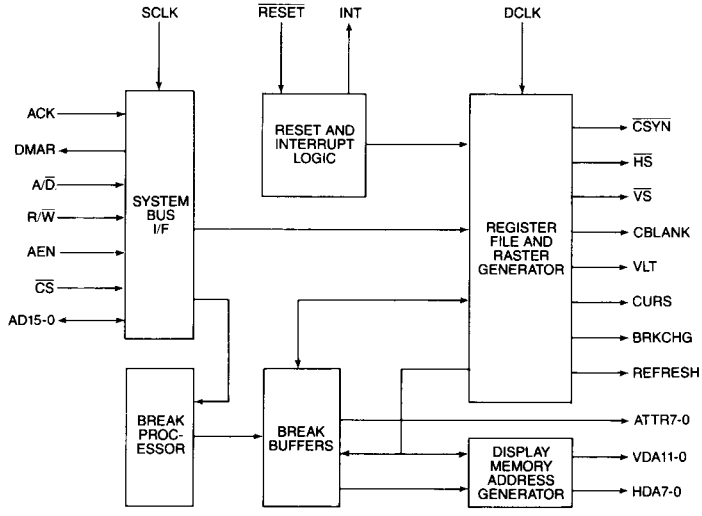
GENERAL DESCRIPTION

The CRT97C11 VIdeo Engine for Windows (VIEW) is a 3rd generation CRT controller following the CRT50x7 family (1st generation) and the CRT9007 (2nd generation). The VIEW is designed to support both bit-mapped graphics and alphanumeric types of CRT displays. This device allows real time manipulation of independent, overlapping windows on the screen with a minimum of processor intervention.

The VIEW architecture provides the system designer with a very high performance and extremely flexible method for supporting the generation of windows on screen. The performance advantage over designs which manipulate windows via software is significant. Windows on screen are defined via a window list maintained in system memory by the system microprocessor and accessed by the VIEW chip. The VIEW chip is also able to access display memory data via a separate memory bus which can be as wide as 24 bits providing a 16M word address range.

The VIEW generates display memory addresses by correlating the X and Y screen coordinates at which the windows start and end with the X and Y display memory addresses. The VIEW stores this information in its three internal break buffers. As the VIEW generates the display memory addresses for the windows, it automatically resolves priority conflicts that arise when two or more windows overlap.

Control of window position on screen, its size and priority relative to other windows and the visible contents of the window are all accomplished via the simple manipulation of data in the window list in system memory. The VIEW buffers and outputs general purpose attribute bits for each window as it generates that window's display memory addresses. The VIEW automatically generates dynamic RAM refresh addresses during the horizontal and vertical retrace intervals.



VIEW FUNCTIONAL BLOCK DIAGRAM

DESCRIPTION OF PIN FUNCTIONS

DISPLAY MEMORY BUS SIGNALS			
PIN NO.	NAME	SYMBOL	FUNCTION
55-48	Horizontal Display Memory Address	HDA7-0	Output. Eight bits of the Display Memory Address corresponding to the X portion of the window's data address in display memory. HDA7 is the MSB and HDA0 is the LSB.
60 62-68 1-4	Vertical Display Memory Address	VDA11-0	Output. Twelve bits of the Display Memory Address corresponding to the Y portion of the window's data address in display memory. VDA11 is the MSB and VDA0 is the LSB.
42	Display Clock	DCLK	Input. This signal is used to generate all Display Memory Bus cycles. A minimum high voltage of 4.0V must be reached for proper operation.
40	Visible Line Time	VLT	Output. This signal is active high for the visible (non-blanked) portion of every horizontal period including the vertical retrace period.
8-5 58-59 57-56	Attributes	ATTR3-0 ATTR5-4/ VDA13-12 ATTR7-6/ HDA9-8	Outputs. Four of these outputs (ATT3-0) are used as general purpose attributes which are unique to each window being displayed. The other four outputs (ATT7-4) are used as either four additional attribute outputs or as address extension bits for the Horizontal Display Memory Address (ATT7-6/HDA9-8) or the Vertical Display Memory Address (ATT5-4/VDA13-12). The functions served by ATT7-4 are determined by the contents of the Interrupt Enable/Mode Register (R17[1,0]).
41	Refresh	RFRSH	Output. This signal is active high when the VIEW is generating dynamic RAM refresh addresses for the display memory. The refresh address is output on the Vertical Display Memory Address bus during the portion of the horizontal period in which the video is blanked (when VLT is low).
47	Break Change	BRKCHG	Output. The active high state of this signal indicates that the next Display Clock generates a new visible horizontal break.

VIDEO DRIVE SIGNALS			
PIN NO.	NAME	SYMBOL	FUNCTION
46	Horizontal Sync	$\overline{HS}$	Input/Output. This signal initiates a horizontal retrace. Its position and pulse width are programmable. After a hardware or software reset this signal will behave as an input. Programming this signal as an input allows the horizontal scan rate to be synchronized to an external source. An external pullup resistor will be required to guarantee that this signal is inactive high after power-up.
45	Vertical Sync	$\overline{VS}$	Input/Output. This signal initiates a vertical retrace. Its position and pulse width are programmable. After a hardware and software reset this signal will behave as an input. Programming this signal as an input allows the vertical scan rate to be synchronized to an external source. An external pullup resistor will be required to guarantee that this signal is inactive high after power-up.

VIDEO DRIVE SIGNALS			
PIN NO.	NAME	SYMBOL	FUNCTION
39	Composite Blank	CBLANK	Output. This signal goes active high when a vertical or horizontal retrace is going to be initiated. The signal stays active for the entire retrace. CBLANK is used to blank the video to the CRT. The CBLANK signal can be skewed 0 to 3 DCLK's with respect to the Display Memory Address and Attribute bus timings.
38	Composite Sync	$\overline{\text{CSYN}}$	Output. This signal provides a true RS-170 composite sync waveform with equalization pulses and vertical serrations in both interlace and non-interlace formats. Figure 14 under Functional Description illustrates the $\overline{\text{CSYN}}$ output in both modes.
36	Cursor	CURS	Output. This signal is active high for the programmed number of DCLK periods on each of the programmed scan lines (see descriptions of R11 thru R15). The CURS output can be skewed 0 to 3 DCLK's from the Display Memory Address and Attribute bus timing.
SYSTEM BUS SIGNALS			
11-26	System Address/Data Bus	AD15-0	Input/Output. These 16 signals are used by the system processor to access the internal registers of the VIEW chip when it is in the peripheral mode. When in the master mode then the VIEW chip controls the bus in order to access the window list in system memory. Following a hardware or software reset the VIEW chip will be in the peripheral mode.
28	Chip Select	$\overline{\text{CS}}$	Input. If VIEW is not performing a DMA cycle, the active low state of this input will allow the system to clock data into or read data out of the internal VIEW registers while the inactive high state of this signal will force AD15-0 into an input (high impedance) state. A minimum high voltage of 4.0V must be output for proper operation.
31	Address Enable	AEN	Input. When the VIEW is operating as bus master and performing a DMA cycle, the active high state of this signal will enable the VIEW to output the System Memory Address on AD15-0. The VIEW is not affected by the state of this input unless it has received ACK and is performing a DMA cycle.
30	Read/Write	R/$\overline{\text{W}}$	Input. This signal is used to qualify the $\overline{\text{CS}}$ input and determines whether the system is writing data into (low state) or reading data from (high state) the VIEW registers.
33	DMA Request	DMAR	Output. This signal is driven active high by VIEW to request use of the System Address/Data bus. It will only become active if the DMA Acknowledge (ACK) input is inactive. It remains active high throughout the entire DMA operation.
35	DMA Acknowledge	ACK	Input. This active high signal acknowledges a DMA request. It is used to enable the VIEW's DMA mechanism. The system should drive ACK inactive low after DMAR is negated. The state of this signal will not affect VIEW operation unless DMAR is being driven active high.
37	Interrupt	INT	Output. This signal is driven active high when VIEW encounters an enabled interrupt causing condition. This output is reset by reading the Interrupt Status register or a hardware reset. NOTE: This signal should be connected to a level sensitive interrupt input as no edge can be guaranteed between successive interrupts. This output will be reset one full DCLK period after $\overline{\text{CS}}$ goes high after an Interrupt Status 1 register read.
34	System Clock	SCLK	Input. This signal is used to generate all system DMA bus cycles. The rising edge of this signal is used by the VIEW to clock in data from the system memory during a DMA cycle. This signal may be "stretched" by the system to generate long read cycles. A minimum high voltage of 4.0V must be output for proper operation.
29	Address/Data	A/$\overline{\text{D}}$	Input. When the VIEW is in the peripheral mode, this signal selects whether a system bus access is to the Register Pointer Latch or to the register file.
32	Reset	RESET	Input. This active low signal puts the VIEW in a known, inactive state and resets all raster counters. Activating this input has the same effect as executing the software Reset command. The following VIEW signals will be left in the indicated states: $\overline{\text{VS}}$ —Input $\overline{\text{HS}}$ —Input $\overline{\text{VLT}}$ —Inactive Low $\overline{\text{CSYN}}$ —Inactive High CURS —Inactive Low RFRSH —Inactive Low AD15-0 —Inputs DMAR —Inactive Low INT —Inactive Low CBLANK—Active High

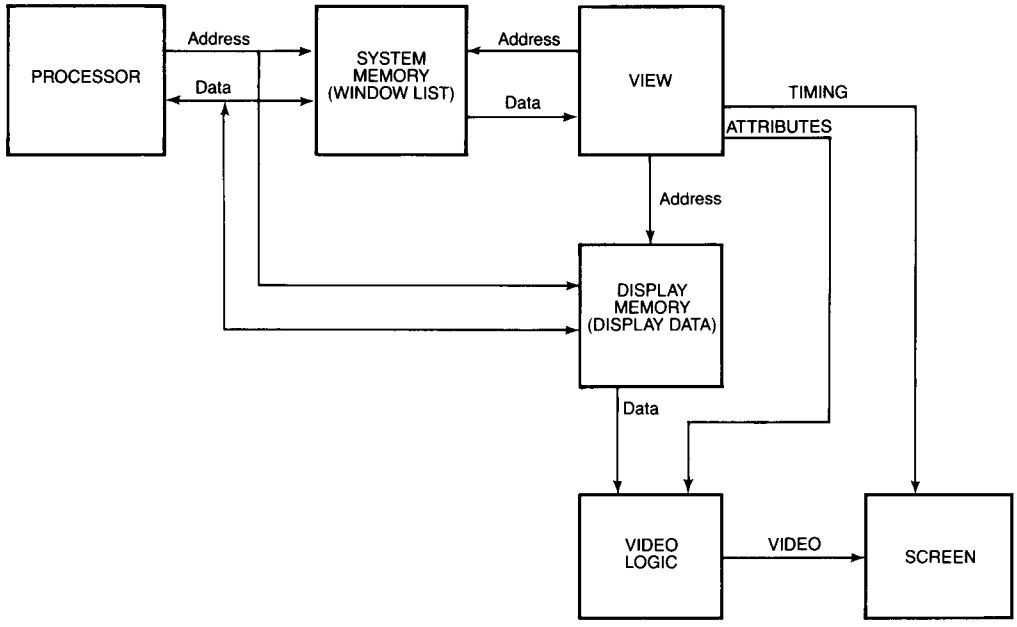


FIGURE 1—SYSTEM ELEMENTS

FUNCTIONAL DESCRIPTION

SYSTEM FUNCTIONS

A display system based on the VIEW must have a minimum set of elements as described in the following sections. The block diagram in Figure 1 provides a simple conceptual illustration of the interconnection of these elements. The sections below will define the purpose of these major elements and also describe how they interact with each other. The VIEW does not restrict the architecture chosen in any particular way and choices made will be determined by the overall system performance goals and cost guidelines.

System Elements

The six blocks shown in Figure 1 are described below in more detail. The interconnection of the blocks are depicted in terms of the address/data buses or other primary inputs or outputs. The details of the interconnection will be determined by the memory arbitration methods implemented and are not being shown here.

a) Processor—The processor can be a system microprocessor (such as the 8088, 80X86, 680X0, etc.) that is responsible for all system functions or a microprocessor dedicated to handling all display and drawing related functions. The processor is responsible for maintaining the window list in system memory and the display data in display memory. For bit mapped graphics applications, overall performance could be improved if a special purpose coprocessor is added to handle the drawing related functions.

b) Window List—The window list is maintained in system memory by the processor. For a definition of the window list and its contents see the section on Screen Management below. The VIEW accesses the window list in order to determine the display memory address sequencing required to generate the windowing effect on screen.

c) VIEW—The VIEW accesses data from the window list, processes it and stores it in internal buffers. The VIEW chip performs all display memory addressing in accordance with the data accessed from the window list. The VIEW can perform both window list and display memory access simultaneously. In addition the VIEW generates video synchronization signals which define the overall video format.

d) Display Data—The VIEW outputs addresses to display memory in a sequence defined by the window list which automatically takes into consideration the overlapping of windows. Display data is stored in memory by the processor in individually allocated areas for each image. Display data can consist of either bit-mapped images or ASCII-based character data. Bit-mapped images may be either single-plane monochrome or multi-plane color. Contention between the VIEW and the processor for access to display memory must be supported at the system design level.

e) Video Logic—This block performs all operations necessary on the data output from display memory before being output as a video signal to the screen. If the data represents bit-mapped images then some typical operations might be serial-to-parallel conversion, color look-up and conversion to RGB output, and other shift operations that may be

required to move images within a window. If the data represents ASCII characters in an alphanumeric only system then some typical functions would include a character generator and attributes controller. In addition the VIEW can output up to 8 attribute signals which are unique to each window and may be used to control specific visual effects.

f) Screen—The output display device can be a CRT or flat panel screen. The VIEW provides all the timing signals required to synchronize the display. See Video Output section below for a more detailed description of these outputs.

System Interaction and Timing

In order to address the problem of memory contention between the processor, VIEW, system memory and display memory, an understanding of the interaction between these elements and their relative timing is important. Whereas the processor can access only one memory (system or display) at a time (unless two processors are used in the system design), the VIEW is capable of performing both memory accesses simultaneously. Refer to Figure 1 while reading this section.

Processor and system memory—Updates of the window list will require activity on the bus between the processor and system memory. This can occur at any time due to operator input or commands received over communications networks which are asynchronous to the screen refresh process. Resolving contention between the processor and the VIEW for access to system memory can make use of the DMA handshake signals provided by the VIEW (see the section on System Interfaces below) or through the use of dual port RAM's.

VIEW and system memory—VIEW accesses the window list whenever it needs to fill its break buffers. This is a periodic process only with reference to the refresh rate. During a given frame refresh the actual timing of these events is determined primarily by a Break Processing Delay parameter and the spacing and number of vertical breaks on the screen. These events always occur before data is needed for screen refresh. See the section on Break Processing below.

Processor and display memory—As in the case with activity between the processor and system memory, update of display data is also driven by commands received via operator input or over communications networks. These events are asynchronous to the screen refresh process. Support for resolving memory contention must be implemented in external logic.

VIEW and display memory—VIEW accesses display memory for the purpose of refreshing the screen image. Therefore this activity is completely synchronous with the video timing and is normally considered the highest priority process within the system.

DISPLAY OUTPUT AND STORAGE

The window list in system memory is used to define and manage windows on screen and specify the data displayed within the windows. The following sections describe the parameters used and how screens and display memory are organized.

Screen Management

The positioning and sizing of windows on a display screen is defined through the use of X and Y coordinates (0, 0 is defined as the upper left corner of the screen). The maximum values of these coordinates is determined by the screen resolution which is defined by the contents of the timing registers (see Operational Description). The absolute maximum values, as determined by the sizes of the coordinate fields in the window list, are 256 for the X coordinate and 4096 for the Y coordinate. Figure 2 shows a screen with a single window in place. The actual location of a window on the screen is determined by a set of 4 coordinates. The left and right edges of the window are defined by XSTART and XEND coordinates and the top and bottom edges of the window are defined by YSTART and YEND coordinates. The width and height of the window is automatically determined from these coordinates. The XSTART and XEND coordinates are called horizontal breaks and the YSTART and YEND coordinates are called vertical breaks (see Break/Link Concept below). These coordinates are stored in the window list data for each window. Horizontally, the spacing of each coordinate is determined by the display data bus width, and vertically, the spacing of each coordinate is one pixel (see the next two sections on Display Memory Management and the Window List).

Display Memory Management

The images viewed through windows on the display screen are determined by a two segment address. This address points to a display memory location which represents the upper left hand corner of the image that appears in the window. The two segment address consists of an X MEMORY ADDRESS and a Y MEMORY ADDRESS. The amount of display memory required is dependent on the application and is a function of the number of independent images to be maintained in memory at the same time and the image sizes to be stored. The VIEW limits the X and Y addressing ranges to 8 bits and 12 bits respectively. Each address segment can be extended by 2 bits. Figure 3a shows a rectangular view of display memory with the unextended limits of addressing indicated. An image area is defined that occupies memory from addresses (0,0) through (X1,Y1). In order to determine what portion of this image will be shown in a window on the screen, an address pointer (X2,Y2) is defined that identifies the upper left corner of the image data to be displayed. This pointer is the X MEMORY ADDRESS and Y MEMORY ADDRESS data stored in the window list. As shown in Figure 3b, each memory location stores N bits of data which is determined by the display memory data bus width.

Window List

The window list (described in detail below under Window Management) contains 6 fields of information that define the relationship between the window on screen and the portion of the display memory image that is shown in the window. As described above, four of the fields are XSTART, XEND, YSTART and YEND which define the position and size of the window on screen. A display memory location (X MEMORY ADDRESS and Y MEMORY ADDRESS) is defined for the upper left corner of the window and all other memory locations are mapped in direct correlation to the width and height of the window. Figure 4 illustrates this relationship. The example shows that the location of the window on screen and the location of the image in memory are independent of each other. The VIEW uses the X MEMORY ADDRESS and the Y MEMORY ADDRESS plus the width and height (ΔX, ΔY) to determine the range of display memory addresses to be generated. Figure 4 assumes N = 8 for the display data bus width which results in a screen size of 1K x 1K pixels and a window size of 256 x 256 pixels.

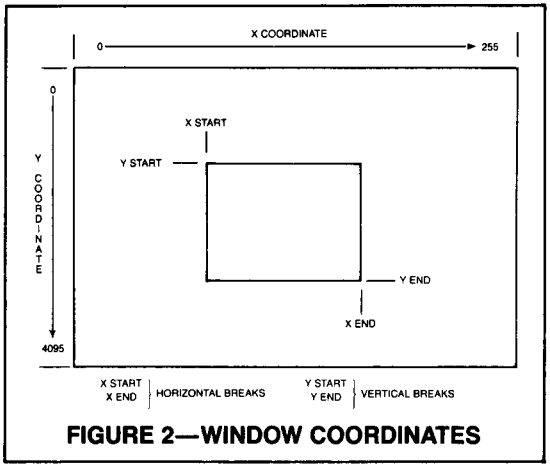


FIGURE 2—WINDOW COORDINATES

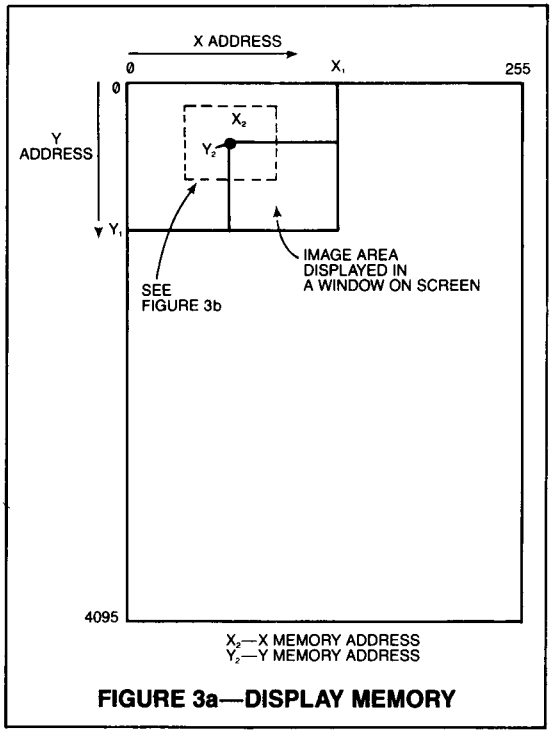


FIGURE 3a—DISPLAY MEMORY

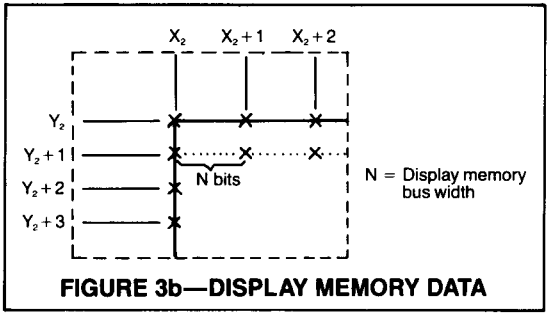


FIGURE 3b—DISPLAY MEMORY DATA

WINDOW MANAGEMENT

The window list is a contiguous set of 16 word blocks of memory which define all parameters associated with each window. A maximum of 127 windows can be defined in a given window list. Up to 32 independent window lists can be maintained in memory at the same time (see R16[7,3] in Operational Description section).

Window List Contents

Each window requires the definition of the following parameters (see Figure 5 for the format):

W0: D15-D12—General Purpose Attribute Bits

The four attribute bits, D15 thru D12, are output on the ATTR3 thru ATTR0 pins. These signals are general purpose and are active during the time the VIEW is generating display memory addresses for the window to which these attributes are assigned.

W0: D10—Background Window Tag Bit

If this bit is set to a one, the associated window is displayed as a background window. The VIEW will generate the same display memory address for every memory access in the window. The address generated is that specified for the X and Y MEMORY ADDRESS described below. If this bit is set to a zero, then all addresses for this window are generated in the normal incrementing manner.

W0: D9-D8—General Purpose Attribute/X Address Extension Bits

Depending on the programming of R17[0] these bits are either General Purpose Attribute bits or X Address Extension bits. When they are programmed to be X Address Extension bits, they function as the X Memory Address MSB's (D9 is the MSB and D8 is the LSB). When they are General Purpose Attributes, D9 is output on ATTR7 and D8 is output on ATTR6.

W0: D7-D0—X Memory Address

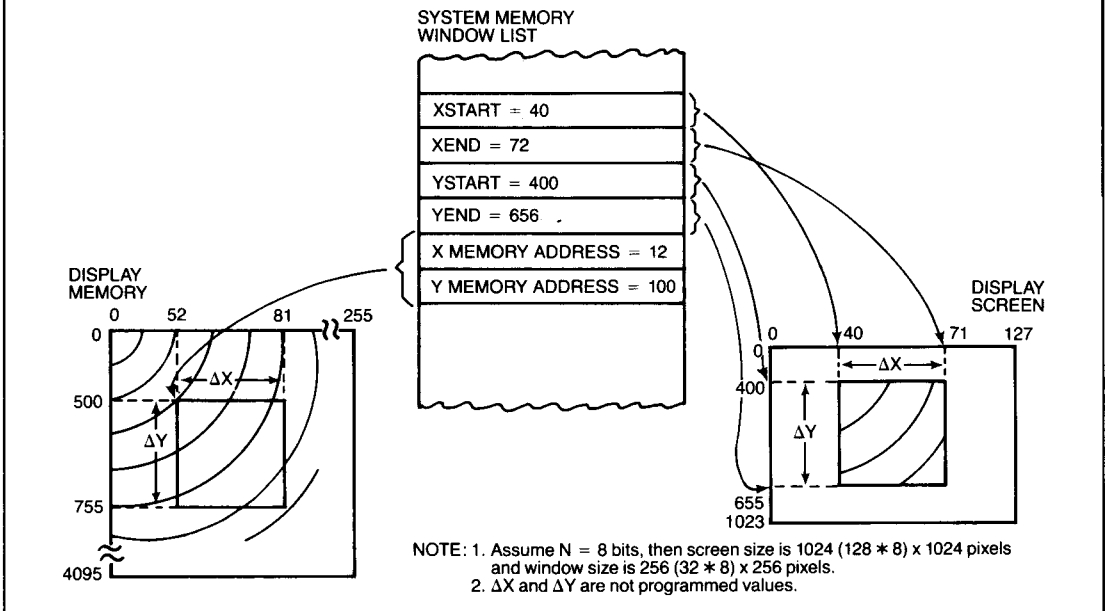
These 8 bits correspond to the X portion of the window's beginning address in display memory. D7 is the MSB and D0 is the LSB. The actual value that is stored in this field depends on the type of window being displayed (normal or background). For a normal window a value equal to [X Memory Address—X Start Break] should be stored (see Figure 4) and for background windows the value stored should be equal to [X Memory Address]. Note that if the X Address Extension is used, then the calculation should be performed using the 10-bit value for X Memory Address and both this field and the X Address Extension field should be programmed with the result. Negative results should be represented in 2's complement form.

W1: D13-D12—General Purpose Attribute/Y Address Extension Bits

Depending on the programming of R17[1] these bits are either General Purpose Attribute bits or Y Address Extension bits. When they are programmed to be Y Address Extension bits, they function as the Y Memory Address MSB's (D13 is the MSB and D12 is the LSB). When they are General Purpose Attributes, D13 is output on ATTR5 and D12 is output on ATTR4.

W1: D11-D0—Y Memory Address

These 12 bits correspond to the Y portion of the window's beginning address in display memory. D11 is the MSB and D0 is the LSB. The actual value that is stored in this field depends on the type of window being displayed (normal or background). For a normal window a value equal to [Y Memory Address—Y Start Break] should be stored (see Figure 4) and for background windows the value stored should be equal



NOTE: 1. Assume N = 8 bits, then screen size is 1024 (128 * 8) x 1024 pixels and window size is 256 (32 * 8) x 256 pixels.
2. ΔX and ΔY are not programmed values.

FIGURE 4—MAPPING MEMORY TO SCREEN

	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
W0	AT3	AT2	AT1	AT0	0	BK	X EXT/AT7,6		X MEMORY ADDRESS							
W1	0	0	Y EXT/AT5,4		Y MEMORY ADDRESS											
W2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W3	0	INT	0	0	Y START BREAK											
W4	0	WINDOW #							0	0	0	PRIORITY				
W5	X	X	X	X	X	X	X	X	Y LINK							S/E
W6	0	WINDOW #							X START BREAK							
W7	X	X	X	X	X	X	X	X	X LINK							S/E
W8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W11	0	INT	0	0	Y END BREAK											
W12	0	WINDOW #							0	0	0	PRIORITY				
W13	X	X	X	X	X	X	X	X	Y LINK							S/E
W14	0	WINDOW #							X END BREAK							
W15	X	X	X	X	X	X	X	X	X LINK							S/E

FIGURE 5—FORMAT FOR WINDOW LIST

to [Y Memory Address]. Note that if the Y Address Extension is used, then the calculation should be performed using the 14-bit value for Y Memory Address and both this field and the Y Address Extension field should be programmed with the result. Negative results should be represented in 2's complement form.

W3: D14, W11: D14—Window Display Interrupt Tag Bit
If this bit is a one, the VIEW will generate an interrupt (if enabled) during the blanking interval preceding the scan line defined as the YSTART break or the YEND break. W3:D14 provides an interrupt prior to the start of a window on screen and W11:D14 provides an interrupt after the end of a window on screen.

W3: D11-D0, W11: D11-D0—Y Start/End Break Coordinate
This 12-bit value defines a vertical coordinate which corresponds to the scan line on which a window starts or ends on screen. W3: D11-D0 points to the start of a window (YSTART = 400 in Figure 4), while W11: D11-D0 points to the scan line after the end of a window (YEND = 656 in Figure 4). D11 is the MSB and D0 is the LSB. The first break at the top of the screen is 00H and the last is FFFH (assuming the full 12 bit coordinate range is used). With a maximum value of FFFH, the last displayable scan line for a window on screen would be FFEH.

W4: D14-D8, W6: D14-D8, W12: D14-D8, W14: D14-D8—Window Number

This 7-bit value corresponds to the break's window number and represents an offset into the full window list at which all data relating to this window can be found. D14 is the MSB and D8 is the LSB. The contents of these four fields should be the same. Window number 00H is reserved. See the section entitled Window List Addressing for more details.

W4: D4-D0, W12: D4-D0—Priority

This 5-bit value represents this window's priority relative to the other windows. This value is used by VIEW to resolve which of two or more overlapping windows will be displayed on a coordinate by coordinate basis. The contents of these two fields should be the same. D4 is the MSB and D0 is the LSB. NOTE: Two windows with the same priority cannot exist on the same scan line. This presents the restriction that no more than 32 windows can overlap.

W5: D15-D8, W7: D15-D8, W13: D15-D8, W15: D15-D8—Backward Link

This 8-bit value (marked by X's in Figure 5) is not used by the VIEW and therefore could be used by the system software for the purpose of defining a Backward Link. The format of this parameter could be similar to that of the X/Y Link and Start/End Tag parameters defined below (8 bits are reserved) or any other format dictated by the system software. These bits are reserved for this purpose and future versions of the VIEW will not make use of them.

W5: D7-D1, W7: D7-D1, W13: D7-D1, W15: D7-D1—X/Y Link

This 7-bit value is a pointer to the next break in ascending coordinate sequence. The X/Y link is actually a window number which represents the next window encountered when sequencing through the windows along the coordinate axis. This value when combined with the associated Start/End Tag bit uniquely identifies the next break. D7 is the MSB and D1 is the LSB. See section on Break/Link Concept for more detail.

W5: D0, W7: D0, W13: D0, W15: D0—Start/End Tag Bit

This one bit value indicates whether the window break defined in the associated X/Y Link is the start or the end of the window. A value of "1" indicates the end of a window and a value of "0" indicates the start of the window.

W6: D7-D0, W14: D7-D0—X Start/End Break Coordinate

This 8-bit value defines a horizontal coordinate which corresponds to the column at which a window starts or ends on screen. W6: D7-D0 points to the start of a window (XSTART = 40 in Figure 4) while W14: D7-D0 points to the memory word after the end of a window (XEND = 72 in Figure 4). D7 is the MSB and D0 is the LSB. The first break at the left of the screen is 00H and the last is FFH. With a maximum value of FFH, the last displayable memory word for a window on screen would be FEH.

NOTE: The contents of words W2, W8, W9 and W10 are not used by the VIEW and may be utilized by the system processor to store other window related information.

Break/Link Concept

The VIEW accesses the window list in order to determine the exact sequence of display memory addresses to be generated to create the desired windowing effect on screen. The VIEW makes use of both break and link information contained in the window list in order to traverse the list in the correct sequence. As the VIEW traverses the list, it checks the priority and location of each window to determine whether it is active for a given region of the screen. A region is defined as that portion of a screen between two vertical breaks. For each active window the VIEW accesses the address information found in the window list, performs calculations on it and stores it in one of three internal break buffers. The contents of the break buffers are then accessed sequentially to generate the display memory addresses required during the screen refresh period.

Breaks—Breaks are the screen coordinates, both vertical and horizontal, at which windows start or end. Figure 6 provides an illustration. Four windows are shown with the X START and END COORDINATES (horizontal breaks) identified at the top—0, 5, 9, 23, 35, 42, 60 and 100. The Y START and END COORDINATES (vertical breaks) are identified at the left—0, 70, 130, 290, 360, 440, 500.

Links—When processing the window list, the VIEW makes use of a series of pointers that link the breaks together in sequential order. The window list contains two linked lists—one for vertical breaks and one for horizontal breaks. The link parameter contains two data items—a window number and a tag bit. The

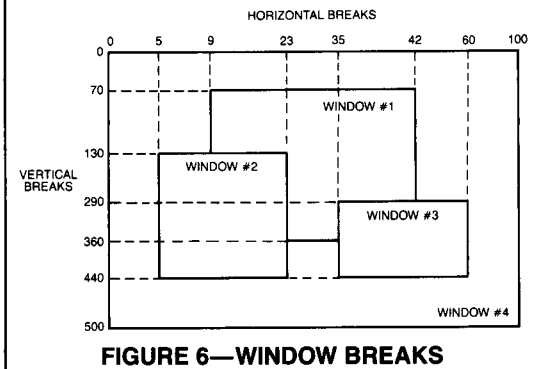


FIGURE 6—WINDOW BREAKS

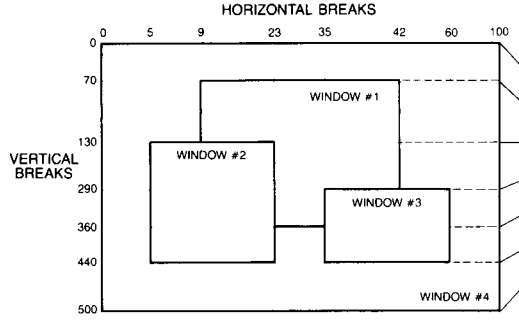
	X START		X END		Y START		Y END	
	COORD	LINK	COORD	LINK	COORD	LINK	COORD	LINK
WINDOW #1	9	2E	43	3E	70	2S	361	3E
WINDOW #2	5	1S	24	3S	130	3S	441	4E
WINDOW #3	35	1E	61	4E	290	1E	441	2E
WINDOW #4	0	2S	101	00	0	1S	501	00

FIGURE 7—WINDOW LINKS

window number identifies which window is encountered next when sequencing through the breaks. The tag bit identifies whether it is the start ("S") or end ("E") of the window. (See the section under Window List Contents for a definition of these data items.) Figure 7 lists all the breaks and links for the example shown in Figure 6. End breaks in Figure 6 represent actual coordinates of last scan line or last display memory word for each window while in Figure 7 each end break value stored in the window list is one more than defined in Figure 6. The VIEW starts processing breaks at the upper left corner of the screen (see Break Processing below), which in this example is window 4. The following list describes the breaks and links along the horizontal axis for this example.

- Break "0"—Coordinate for start of window 4.
- Link "2S"—Pointer to the start of window 2.
- Break "5"—Coordinate for start of window 2.
- Link "1S"—Pointer to the start of window 1.
- Break "9"—Coordinate for start of window 1.
- Link "2E"—Pointer to the end of window 2.
- Break "24"—Coordinate for end of window 2.
- Link "3S"—Pointer to the start of window 3.
- Break "35"—Coordinate for start of window 3.
- Link "1E"—Pointer to the end of window 1.
- Break "43"—Coordinate for end of window 1.
- Link "3E"—Pointer to the end of window 3.
- Break "61"—Coordinate for end of window 3.
- Link "4E"—Pointer to the end of window 4.
- Break "101"—Coordinate for end of window 4.
- Link "00"—Terminator for end of horizontal breaks.

As shown, a special link, or terminator, equal to "00" is used to indicate the last break in the sequence. A similar sequence exists for the vertical breaks. The VIEW uses the two independently linked lists to perform the break processing described in the next section. It is the responsibility of the window management software to create and maintain these links in the window list in system memory when manipulating windows on screen.



REGION	Y COORD	ACTIVE WINDOWS	BREAK BUFFER
1	0-69	4	1
2	70-129	1, 4	2
3	130-289	1, 2, 4	3
4	290-360	1, 2, 3, 4	1
5	361-440	2, 3, 4	2
6	441-500	4	3

FIGURE 8—BREAK PROCESSING EXAMPLE

	FRAME REFRESH	BUFFER USED	BREAK PROCESSING	BUFFER USED	ACCESS TIME
			REGION 6	3	39 SCLK's
FRAME REFRESH			① LAST BREAK PROCESSED		10 SCLK's
			BREAK DELAY (R10)		
VERTICAL RETRACE			② REGION 1	1	44 SCLK's
			REGION 2 ③	2	39 SCLK's
			REGION 3	3	39 SCLK's
		④ 0			
FRAME REFRESH	REGION 1	1 70	REGION 4 ⑤	1	39 SCLK's
	REGION 2	2 130	REGION 5 ⑥	2	42 SCLK's
	REGION 3	3 290	REGION 6 ⑥	3	39 SCLK's
	REGION 4	1 360			
	REGION 5	2 440	⑦ LAST BREAK PROCESSED ⑥		10 SCLK's
	REGION 6	3 500	BREAK DELAY (R10)		
			REGION 1	1	39 SCLK's

FIGURE 9—BREAK PROCESSING TIMING

Break Processing

For each screen refresh, the VIEW accesses the window list by reading all vertical breaks for each region of the screen (more than one vertical break in a region is possible if multiple windows either end or start on the same scan line) and then reading all horizontal breaks. When reading the horizontal breaks, the VIEW processes the address and attribute data for each active window and stores this data in the next available break buffer.

Figure 8 and 9 will be used to illustrate this process. Figure 8 repeats the same screen layout as shown in Figure 6 and adds four columns which identify the region, the range of Y coordinates for that region, active windows in that region, and the break buffer used to store data for that region. Figure 9 illustrates the timing of the window list accesses and the break buffer activity for a single frame refresh period.

During a given frame refresh, the Last Break Processed interrupt will occur when the vertical break for the end of the screen is processed ①. At this time the VIEW will not attempt to start break processing for the next frame refresh until after the Start Delay (see R10[5,0]) timeout has terminated ②. This timeout may occur before or after the start of vertical retrace and is strictly a function of the Start Delay parameter and when the Last break Processed interrupt occurs. This delay allows time for the processor to manipulate the window list. At this time the VIEW will request access to the system bus by activating the DMAR signal. When the ACK signal is activated, the VIEW will process the first four vertical breaks which provides all the data needed to refresh these regions on screen ③. When frame refresh starts, the VIEW will make use of the data in break buffer 1 to refresh the first region on screen ④. At the completion of this period, the contents of break buffer 1 are no longer needed and the VIEW processes the next vertical break (between vertical coordinates 29 and 36—region 4). The data for refreshing region 4 is now placed in break buffer 1 ⑤. Similarly, at the completion of the refresh of regions 2, 3 and 4, break buffers 2, 3 and 1 become available and the VIEW processes the breaks for regions 5 and 6 and stores this data in break buffers 2 and 3 and then processes the last break, represented by the Y break at coordinate 500 (no break buffer is required in this case) ⑥. Once again the VIEW will generate the Last Break Processed interrupt and the entire cycle will start again ⑦.

Figure 9 also shows the amount of time that the VIEW takes to perform the break processing. This time is measured in SCLK's. The formula to calculate this timing is as follows—

$$8W + 3Y + 4$$

where W = Number of total windows (for the example in Figure 6, W would equal 4 for all regions).

Y = Number of vertical breaks with the same break coordinate (start or end) for a given region. For the example in Figure 6, Y equals 1 for regions 1, 2, 3, 4, 6 and Y equals 2 for region 5. Y = 2 for region 5 because windows 2 and 3 end on the same scan line (same Y break coordinate).

Three special conditions exist with respect to the above calcuation—

1) If the break being processed is the final break (break which represents the last scan line on the screen), then the formula is modified as follows—

   $3Y + 7$

2) If the break being processed is the first break (break which represents the first scan line on the screen—Region 1 in the example), then the formula is modified as follows—

   $8W + 3Y + 9$

3) If the break being processed is the first break after the VIEW has been given a Start command, then an additional 2 SCLK's should be added to the result.

Window List Addressing

As indicated above, each window is defined by the parameters stored in a 16 word block of system memory. In addition to the 127 blocks of system memory required to define all possible windows (numbered 1 through 127), there is also a block reserved for window # 0 that serves a special purpose. When the VIEW processes breaks for the next frame refresh, it starts with the block that defines the first window to be encountered at the upper left corner of the screen. The pointer to that block is found in the first two words of the block for window # 0. Figure 10 shows the contents of this special block of data. The LSB of the first word identifies the X START LINK and S/E tag bit and the LSB of the second word identifies the Y START LINK and S/E tag bit. All other bits in this block of memory should be reset. These two link parameters define the entry points to the linked lists for vertical and horizontal breaks.

The address generated by the VIEW for accessing the window list consists of three segments (see Figure 11). The 5 MSB's (AD15-11) are defined by the contents of the Window List Start Address found in register (R16[7,3]). This base address allows the VIEW to access up to 32 independent window lists. The next 7 bits (AD10-4) are defined by the specific window number which is included in the data accessed from the window list. The 4 LSB's (AD3-0) are the offset within each window list which points to a specific data item.

SYSTEM INTERFACES

System Memory

The VIEW operates in two modes—peripheral and master. Following a hardware $\overline{\text{RESET}}$ or a software Reset command, the VIEW will be in peripheral mode. In this mode, the processor can access the VIEW's internal registers by means of the $\overline{\text{CS}}$ and R/$\overline{\text{W}}$ inputs. After a Start command, the VIEW will begin to generate DMA requests. When it receives a DMA acknowledge from the processor, the VIEW will operate in bus master mode (see Figure 12). In this mode, the VIEW will use its DMA circuitry to request the use of the system bus from the permanent master (typically the system processor) when it needs to fill its break buffers. When the VIEW disables its DMA request, it will again revert to peripheral mode (see Figure 13). Table 1 defines the activity on the System Address/Data Bus for the control signal states shown.

After processing the last break for a given frame refresh, the VIEW will wait the period of time specified in the Start Delay register before it starts to access system memory again. After this time, the VIEW will start to load the break buffers for the first break of the next frame refresh. This feature prevents system memory access conflicts as it provides the processor with a window of programmable length in which it can address the VIEW as a peripheral, disable the VIEW's DMA mechanism, access the internal registers and access the window list in system memory.

ADDRESS	WINDOW ϕ DATA															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	X START LINK							S/E[1]
1	0	0	0	0	0	0	0	0	Y START LINK							S/E[1]
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FIGURE 10—WINDOW ϕ CONTENTS

NOTE: [1] Both S/E tag bits should be set equal to zero because the first window boundaries encountered will always be the start of a window.

FIGURE 11—WINDOW LIST ADDRESSING

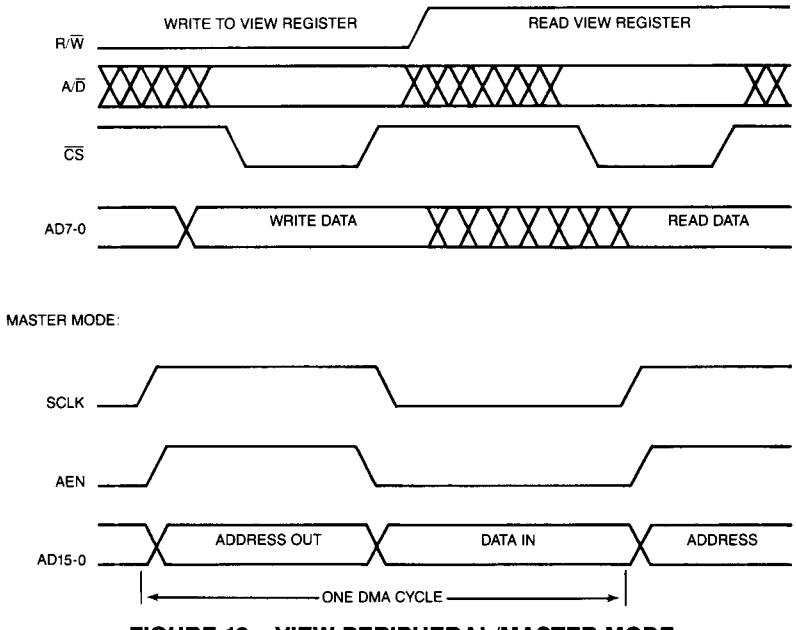


FIGURE 12—VIEW PERIPHERAL/MASTER MODE

The VIEW will tri-state its System Address/Data bus (AD15-0) when it is not performing a DMA cycle and when it is not selected ($\overline{CS}$ in its inactive state) for register access. The processor can also disable the DMA mechanism by issuing a STOP command. In this case the VIEW will not access system memory again until a START command is issued. However, the VIEW will continue to refresh the display memory if transparent refresh is enabled.

Display Memory

The VIEW addresses display memory as a rectangular memory space with a two segment address—8 bits for the horizontal range (HDA7-0) and 12 bits for the vertical range (VDA11-0). Each segment can be independently extended by two bits (see description of General Purpose Attribute Bits below). The logic required to support the RAS/CAS address generation (required when using DRAM's) and to arbitrate display memory accesses must be implemented externally.

If display memory is implemented using DRAM's, then the VIEW supports the refresh of memory through the use of an internal twelve bit binary counter. The outputs of this counter are used to drive the Vertical Display Memory Address bus (VDA11-0) during the portion of the horizontal period in which the video is blanked (when VLT is low). The counter is incremented by the DCLK signal the number of times specified by the Memory Refresh Count register during each horizontal blanking interval (for the entire vertical period). The counter is set to zero (counting is inhibited) by a software Reset command or a hardware RESET. The counter can also be disabled at any time by setting the Memory Refresh Count register to zero. The counter will not initiate the counting sequence until the VIEW receives a Start command. The VIEW drives the RFRSH output active high when it is driving the VDA11-0 bus with the counter output. To take advantage of the VIEW's refresh function, the VDA11-0 bits should be used for the DRAM row address.

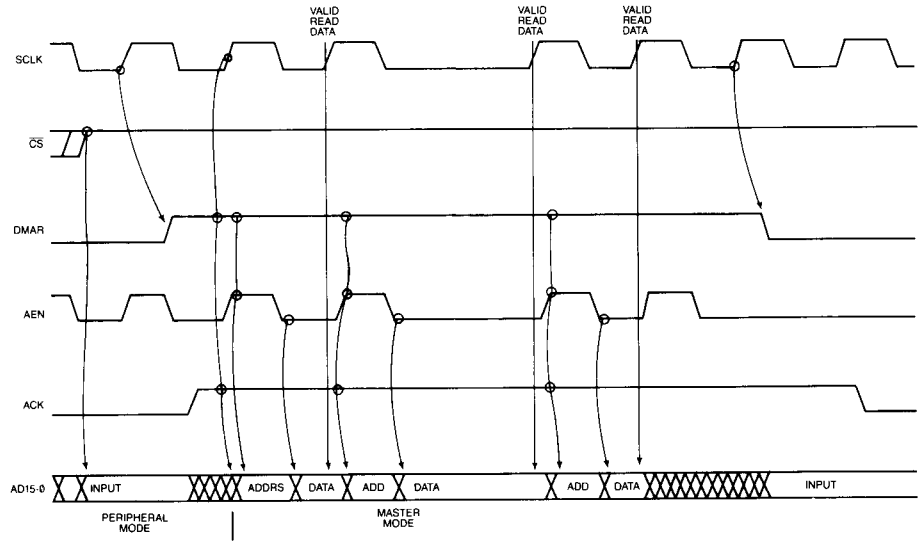


FIGURE 13—VIEW DMA CYCLE

MODE	R/W*	CS*	ADR	DMAR	ACK	AEN	AD15 – AD8	AD7 – AD0
PERIPHERAL	1	0	0	0	X	X	X	READ REG FILE
PERIPHERAL	1	0	1	0	X	X	X	READ REG POINTER
PERIPHERAL	0	0	0	0	X	X	X	WRITE REG FILE
PERIPHERAL	0	0	1	0	X	X	X	WRITE REG POINTER
PERIPHERAL	1	0	0	X	0	X	X	READ REG FILE
PERIPHERAL	1	0	1	X	0	X	X	READ REG POINTER
PERIPHERAL	0	0	0	X	0	X	X	WRITE REG FILE
PERIPHERAL	0	0	1	X	0	X	X	WRITE REG POINTER
PERIPHERAL	X	1	X	0	X	X	HIGH IMPEDANCE (INPUT) STATE	
PERIPHERAL	X	1	X	X	0	X	HIGH IMPEDANCE (INPUT) STATE	
MASTER	X	1	X	1	1	0	HIGH IMPEDANCE (INPUT) STATE	
MASTER	X	1	X	1	1	1	DMA ADDRESS (OUTPUT)	

SYSTEM ADDRESS/DATA BUS
TABLE 1

Video Output

Cursor Control

The VIEW provides a cursor output (CURS—pin 36) with separate programmable X and Y start and duration values. The cursor output will become active when the value programmed in the Horizontal Cursor Position register is equal to the X component of the raster counter and the value programmed in the Vertical Cursor Position register is equal to the Y component of the raster counter. It will remain active for the number of DCLK periods specified in the Horizontal Cursor Duration register before returning to its inactive state. The cursor output's action will be repeated on the scan lines following the one specified in the Vertical Cursor Position register for the number of scan lines specified in the Vertical Cursor Duration register.

As the Horizontal and Vertical Cursor Position registers are changed, the information read from the Interrupt Status register will indicate in which window the cursor resides. This is determined by the location of the upper left corner of the cursor block as specified in the Horizontal and Vertical Cursor Position registers. If the cursor position registers place the cursor simultaneously in two or more overlapping windows, the status will report it as being in the window with the highest priority.

Window Attributes

Unique attributes can be specified for each window. Two types of attributes are used internally by the VIEW—the window interrupt tag bits and a background window tag bit. There are also eight general purpose attribute bits. All of these bits are accessed by the VIEW during break processing and stored internally in break buffers.

Window Interrupt Tag Bits

Two interrupt tag bits are assigned to each window. When either of these bits are set and the Y Break Interrupt Enable bit (R17[4]) is set, an appropriate interrupt will be generated by the VIEW during the horizontal retrace period. The tag bit associated with the Y Start break will cause an interrupt to be generated during the horizontal retrace period prior to the first scan line of the window. The tag bit associated with the Y End Break will cause an interrupt to be generated during the horizontal retrace period following the last scan line of the window. The timing of these interrupts is independent of whether the first or last scan lines of the window are obscured by a higher priority window. See discussion of R17[4] in Register Descriptions section and W3: D14 and W11: D14 in the Window List section.

Background Window Tag Bit

See description of this bit (W0: D10) in the section on the Window List.

General Purpose Attribute Bits

The eight general purpose attribute bits do not affect the internal operations of the VIEW, however they are read and output on the ATTR7-0 pins during the time the VIEW is generating display memory addresses associated with that window. The functionality of four of the eight attribute bits can be programmed by the Attribute Select bits in R17[1,0] (see Register Descriptions section). Independently, two of the bits can be programmed to extend the Horizontal Display Memory Address and the other two can be programmed to extend the Vertical Display Memory Address.

Horizontal and Vertical Sync

The horizontal and vertical synchronization outputs are programmable with respect to their pulse width and position relative to the horizontal and vertical visible display times. See Figure 14 for typical waveforms in interlaced and non-interlaced modes. They can also be configured so as to allow external signals to initiate horizontal and vertical retrace cycles. If the external horizontal sync is enabled and an external signal drives the $\overline{HS}$ input low, the VIEW will initiate a horizontal retrace cycle on the leading edge of the second DCLK pulse following $\overline{HS}$ going low. This will lock the VIEW's horizontal frequency to that of the external signal. If the external vertical sync is enabled and an external signal drives the $\overline{VS}$ input low, the VIEW will initiate a vertical retrace cycle on the leading edge of the next $\overline{HS}$ pulse. This will lock the VIEW's vertical frequency to that of the external signal. When external sync is enabled, the vertical and horizontal periods must be programmed to be the same as the master sync generator. Note that it is possible for the first frame to be out of sync in interlace mode.

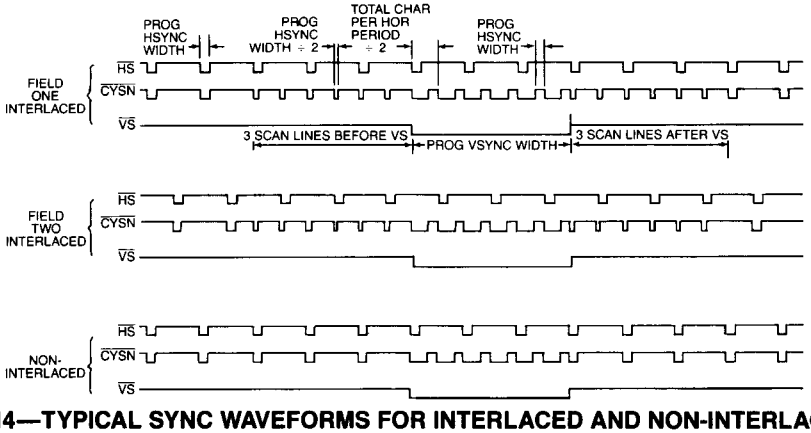


FIGURE 14—TYPICAL SYNC WAVEFORMS FOR INTERLACED AND NON-INTERLACED MODES

OPERATIONAL DESCRIPTION

REGISTER ACCESSIBILITY

Figures 16 and 17 illustrate the bit layout of all addressable registers within the VIEW. Access to these registers takes place over the System Address/Data Bus (AD15-0). Specific registers are selected for reading and writing by means of a Register Pointer Latch. Data transfers to and from the registers takes place over the 8 LSB's of the address/data bus (AD7-AD0, see Table 1). Control over the access to both the registers and the Register Pointer Latch is accomplished via the A/$\overline{D}$, R/$\overline{W}$ and $\overline{CS}$ inputs (see Table 2). These registers should not be accessed during a DMA cycle (DMAR and ACK both active).

A/$\overline{D}$	R/$\overline{W}$	FUNCTION
0	0	Write to register in register file
0	1	Read register in register file
1	0	Write to Register Pointer Latch
1	1	Read from Register Pointer Latch

TABLE 2—ACCESS TO VIEW REGISTERS

The Register Pointer Latch is 8 bits wide (Fig. 15). The five LSB's are used to select the desired register. The three MSB's are used to initiate Start, Stop and Reset commands via software. The allowable combinations of bit settings for the Register Pointer Latch are shown in Fig. 15.

D7	D6	D5	D4 D3 D2 D1 D0	
RESET	STOP	START	REGISTER NUMBER	FUNCTION
1	0	0	X X X X X	RESET
0	1	0	X X X X X	STOP
0	0	1	X X X X X	START
0	0	0	←—— N ——→	ACCESS REGISTER N

X—Don't care N—Register address

FIGURE 15—REGISTER POINTER LATCH

Executing one of the three commands requires the exclusive setting of one of the three MSB's. When accessing a register all three MSB's must be reset. Execution of the commands causes the following actions to take place:

RESET: Writing this command will cause the VIEW to respond exactly as though the Reset pin was strobed. All raster counters will be reset and the signals listed below will be left in the states indicated:

$\overline{VS}$	—Input
$\overline{HS}$	—Input
$\overline{VLT}$	—Inactive Low
$\overline{CSYN}$	—Inactive High
CURS	—Inactive Low
RFRSH	—Inactive Low
AD15-0	—Inputs
DMAR	—Inactive Low
INT	—Inactive Low
CBLANK	—Active High

STOP: Writing this command will cause the VIEW to disable its DMA circuitry and drive its CBLANK signal active high. The VIEW will not initiate a DMA request again or drive the System Address/Data bus until it receives a new Start command. Horizontal and vertical sync generation will not be affected.

START: Writing this command will cause the VIEW to initiate its internal operations by starting its raster counters and filling its break buffers. The VIEW always generates even field raster addresses following a Reset command.

NOTE: A minimum delay of 4 full SCLK periods is required between a Stop, Start, or Reset command or a hardware Reset and writing to the VIEW registers. The clock periods are counted starting with the first rising edge of SCLK following the rising edge of $\overline{CS}$.

	D7	D6	D5	D4	D3	D2	D1	D0
R0	HORIZONTAL CYCLE MSB'S							
R1	HOR CY LSB	HORIZONTAL SYNC WIDTH						
R2	INTER MODE	HORIZONTAL DELAY						
R3	HORIZONTAL VISIBLE							
R4	VERTICAL CYCLE LSB'S						SCAN MODE	EXT SYNC ENABLE
R5	VERTICAL CYCLE MSB'S							
R6	VERTICAL SYNC WIDTH							
R7	VERTICAL DELAY							
R8	VERTICAL VISIBLE LSB'S							
R9	VERTICAL VISIBLE MSB'S					MEMORY REFRESH COUNT		
R10	BLANK SKEW	START DELAY						
R11	CURS SKEW	V CUR DUR MSB	HORIZ CURSOR DURATION					
R12	VERTICAL CURSOR DURATION LSB'S							
R13	HORIZONTAL CURSOR POSITION							
R14	VERT CURSOR POS MSB'S							
R15	VERTICAL CURSOR POSITION LSB'S							
R16	WINDOW LIST START ADDRESS						0	0	0
R17	INTERRUPT ENABLE							ATTR SEL

FIGURE 16—VIEW REGISTERS (WRITE)

	D7	D6	D5	D4	D3	D2	D1	D0
RR17	INTERRUPT STATUS 1							
RR18	INTERRUPT STATUS 2							
RR13	HORIZONTAL CURSOR POSITION							
RR14	VERT CURSOR POS MSB'S							
RR15	VERTICAL CURSOR POSITION LSB'S							

FIGURE 17—VIEW REGISTERS (READ)

REGISTER DESCRIPTIONS

HORIZONTAL TIMING—

HORIZONTAL CYCLE (9 Bits—R0[7,0], R1[7])—Write only.

Defines the horizontal period length (visible and blanking time). This field should be programmed with the total number of DCLK periods in the entire horizontal period minus four times the number of DCLK periods specified for the Horizontal Sync Width (see Figure 18).

HORIZONTAL SYNC WIDTH (7 Bits—R1[6,0])—Write only.

Defines the duration of the Horizontal Sync signal ($\overline{HS}$). This field should be programmed with the number of DCLK periods that compose the horizontal synchronization pulse (see Figure 18). The minimum value is 3.

HORIZONTAL DELAY (7 Bits—R2[6,0])—Write only. Defines the delay between the start of the Horizontal Sync signal and the start of the next visible scan line. This field should be programmed with N – 3 where N is the number of DCLK periods between the beginning

of the $\overline{HS}$ signal and the leading edge of the VLT signal. If this register is programmed with a value that is greater then the horizontal blank interval, the horizontal sync pulse will begin before the horizontal blank interval yielding a negative "front porch" (see Figure 18). The minimum value is 1.

HORIZONTAL VISIBLE (8 Bits—R3[7,0])—Write only. Defines the length of the visible portion of the total horizontal period. This field should be programmed with N – 1 where N is the number of DCLK periods that the display is not blanked during the horizontal period (see Figure 18).

NOTE: VIEW requires that the display must be blanked a minimum of 11 DCLK periods (interlaced display) or 7 DCLK periods (noninterlaced display) in a Horizontal Cycle.

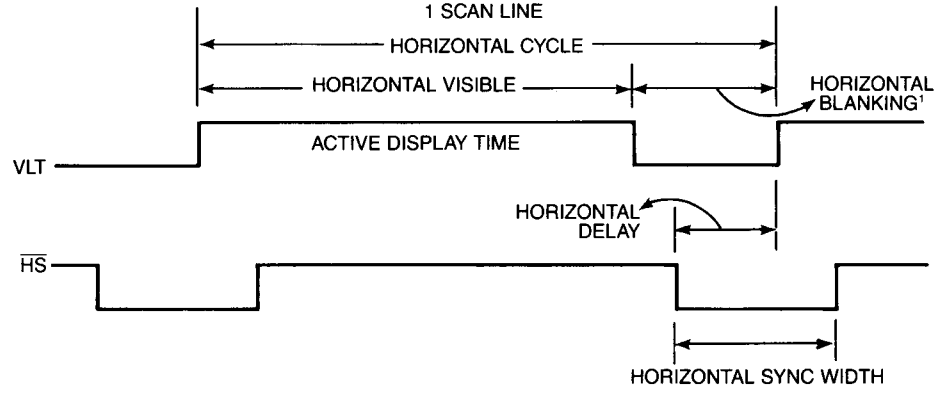
VERTICAL TIMING—

VERTICAL CYCLE (13 Bits—R5[7,0], R4[7,3])—Write only.
Defines the vertical period length (visible and blanking time). This field should be programmed with 6 less than the total number of scan lines in the total vertical period (see Figure 19).

VERTICAL SYNC WIDTH (8 Bits—R6[7,0])—Write only.
Defines the duration of the Vertical Sync signal ($\overline{VS}$). This field should be programmed with the number of horizontal periods (scan lines) that compose the vertical synchronization pulse (see Figure 19).

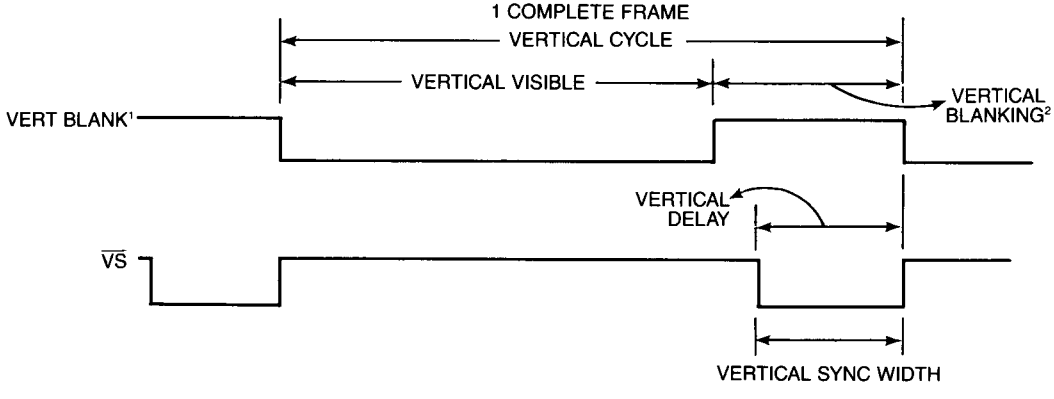
VERTICAL DELAY (8 Bits—R7[7,0])—Write only.
Defines the delay between the beginning of the Vertical Sync signal and the end of the vertical blanking interval. This field should be programmed with N – 2 where N is the number of horizontal periods (scan lines) between the beginning of $\overline{VS}$ and the falling edge of vertical blanking (see Figure 19).

VERTICAL VISIBLE (12 Bits—R9[7,4], R8[7,0])—Write only.
Defines the length of the visible portion of the total vertical period. This field should be programmed with N – 1 where N is the number of horizontal periods (scan lines) that the display is not blanked during the vertical period (see Figure 19).



NOTE: 1. This value is not programmed, however it can be calculated from the other programmed parameters.

FIGURE 18—VIEW HORIZONTAL TIMING



NOTE: 1. This signal not available externally.
2. This value is not programmed, however it can be calculated from the other programmed parameters.

FIGURE 19—VIEW VERTICAL TIMING

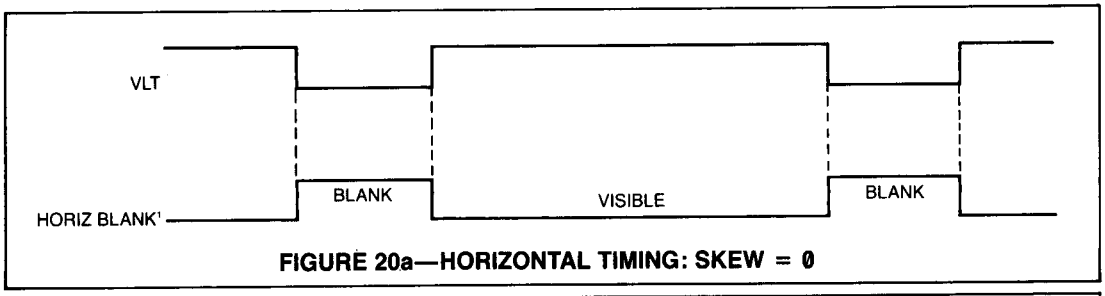
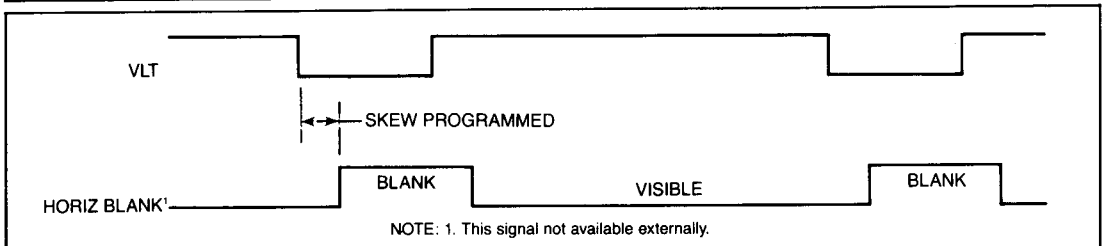


FIGURE 20a—HORIZONTAL TIMING: SKEW = 0



NOTE: 1. This signal not available externally.

FIGURE 20b—HORIZONTAL TIMING: SKEW ≠ 0

CURSOR CONTROL—

HORIZONTAL CURSOR DURATION (5 Bits—R11[4,0])—Write only.
Defines the active time per scan line of the Cursor (CURS) output pulse. This field should be programmed with the number of DCLK periods that the output is active high. The specific scan lines on which the CURS signal will be activated is determined by the contents of the Vertical Cursor Duration and Position fields. This parameter defines the width of the cursor rectangle on the screen.

VERTICAL CURSOR DURATION (9 Bits—R11[5], R12[7,0])—Write only.
Defines the active time per vertical period of the Cursor (CURS) output pulse. This field should be programmed with the number of scan lines that the output is active high. This parameter determines the height of the cursor rectangle on the screen.

HORIZONTAL CURSOR POSITION (8 Bits—R13[7,0], RR13[7,0])—Read/Write.
Defines the absolute horizontal coordinate relative to the visible portion of the screen when the Cursor (CURS) output will go active high. This field should be programmed with the X coordinate of the upper left corner of the cursor rectangle.

VERTICAL CURSOR POSITION (12 Bits—R14[7,4], R15[7,0], RR14[7,4], RR15[7,0])—Read/Write.
Defines the absolute vertical coordinate relative to the visible portion of the screen when the Cursor (CURS) output will go active high. This field should be programmed with the Y coordinate of the upper left corner of the cursor rectangle.

CURSOR SKEW (2 Bits—R11[7,6])—Write Only.
These bits define the number of DCLK periods that the CURS output signal is delayed (skewed) from the Display Memory Address corresponding to the cursor position. If zero skew is specified, the CURS output will be active when the VIEW generates the addresses in direct relation to the cursor position. The maximum cursor skew is three.

MISCELLANEOUS CONTROL—

BLANKING SKEW (2 Bits—R10[7,6])—Write only.
These bits define the number of DCLK periods that the horizontal blank component of the CBLANK signal is delayed (skewed) from the VLT signal as shown in Figure 20b. If, as shown in Figure 20a, zero skew is specified, the edges of the horizontal component of CBLANK will coincide with the edges of VLT. The maximum blanking skew is three.

WINDOW LIST START ADDRESS (5 Bits—R16[7,3])—Write only.
Defines the 5 MSB's of the memory address generated by the VIEW to access the Window List in system memory. This base address allows the VIEW to maintain up to 32 Window Lists in memory at the same time.

SCAN MODE (1 Bit—R4[2])—Write only.
Defines which scan mode will be used by the VIEW for generation of timing signals and Display Memory Addresses. If this bit is reset then non-interlaced mode (even and odd scan lines displayed sequentially in one field) will be used and if set then the interlaced mode (see Interlace Mode register) will be used.

INTERLACE MODE (1 Bit—R2[7])—Write only.
Defines which interlace mode will be used when the Scan Mode bit is set (R4[2] = 1). If this bit is reset (R2[7] = 0) then the normal interlaced mode is enabled which will cause the odd scan lines to be displayed in odd fields and even scan lines to be displayed in even fields. If this bit is set (R2[7] = 1) then the enhanced interlace mode is enabled which will cause the even and odd scan lines to be repeated on successive scan lines for both even and odd fields.

EXTERNAL SYNC ENABLE (2 Bits—R4[1,0])—Write only.
Defines whether sync outputs are generated internally or triggered by external signals. R4[1] defines the source of the Vertical Sync signal (R4[1] = 0 – internal, R4[1] = 1 – external). R4[0] defines the source of the Horizontal Sync signal (R4[0] = 0 – internal, R4[0] = 1 – external).

MEMORY REFRESH COUNT (4 Bits—R9[3,0])—Write only.

Defines how many sequential Display Memory addresses will be generated by VIEW for transparent refresh of dynamic RAM's during the horizontal retrace interval (when the VLT output is inactive low). Bit 3 is the MSB and bit 0 is the LSB. Setting these bits to zero will disable the refresh counter. These bits are reset by a hardware Reset or software Reset command.

START DELAY (6 Bits—R10[5,0])—Write only.

Defines the number of scan lines the VIEW will wait after accessing the Window List in system memory for the last break. The VIEW will not access the Window List again until this count expires at which time the VIEW will begin processing data again for the first break of the next frame refresh. Bit 5 is the MSB and bit 0 is the LSB. See section on Break Processing under Functional Description.

ATTRIBUTE SELECT (2 Bits—R17[1,0])—Write only.

Defines the functionality of the VIEW's ATTR7-4 general purpose Attribute output pins. The programming of these bits will determine whether these outputs function as attribute information or as extension bits to the Display Memory Address bus. See Table 3 for programming of these bits and the Functional Description section for an explanation of functions associated with these outputs.

VIEW OUTPUT	ATTRIBUTE SELECT R17[0] = 1	ATTRIBUTE SELECT R17[0] = 0
ATTR7 ATTR6	X ADDR EXT MSB X ADDR EXT LSB	GP ATTRIBUTE GP ATTRIBUTE
	R17[1] = 1	R17[1] = 0
ATTR5 ATTR4	Y ADDR EXT MSB Y ADDR EXT LSB	GP ATTRIBUTE GP ATTRIBUTE

ATTRIBUTE SELECT PROGRAMMING TABLE 3

INTERRUPT ENABLE (7 Bits—R17[7,5,4,3,2])—Write only.

These bits, when set, enable the VIEW to generate an interrupt when the appropriate conditions exist. These bits are reset to zero by a hardware Reset or a software Reset command. The following defines the specific conditions associated with each of the bits in this register.

Bit 7 (Interrupt Enable)
This bit, when set, will allow the VIEW to drive its INT pin high whenever the Interrupt Pending bit in the Interrupt Status 2 register goes high. When reset, the VIEW cannot drive the INT pin high. This bit is a global interrupt enable and has higher priority than the other enable bits in this register.

Bit 5 (Cursor Window Interrupt Enable)
This bit, when set, will allow the VIEW to set the Interrupt Pending bit after the last break has been processed whenever the cursor position (upper left corner of the cursor area) is moved into another window. When reset, this condition will be ignored.

Bit 4 (Y Break Interrupt Enable)
This bit, when set, will allow the VIEW to set the Interrupt Pending bit whenever a Y break with the Interrupt Tag bit set is encountered when generating Display Memory Addresses. The same interrupt will be generated for both even and odd fields. When reset, this condition will be ignored.

Bit 3 (Last Break Processed Interrupt Enable)
This bit, when set, will allow the VIEW to set the Interrupt Pending bit whenever the VIEW retrieves a Y break with a terminator window number followed by an X break with a terminator window number. When reset, this condition will be ignored.

Bit 2 (Vertical Retrace Interrupt Enable)
This bit, when set, will allow the VIEW to set the Interrupt Pending bit at the start of vertical blanking time. When reset, this condition will be ignored.

INTERRUPT STATUS 1 (5 Bits—RR17[6, 5, 4, 3, 2])—Read only.

These bits act as flags to indicate which condition/s are causing an interrupt. With the exception of bit 6 (Odd/Even status), these bits are reset by reading the Interrupt Status 1 register, a hardware Reset, or a software Reset command. They can only be driven active when they are enabled in the Interrupt Enable register. This register should not be read unless the Interrupt Pending bit in the Interrupt Status 2 register is high or the VIEW has driven its INT output active.

Bit 6 (Odd/Even)
This bit is set when the next field to be displayed is the odd field and is reset when the next field to be displayed is the even field. This status bit becomes valid shortly before the Vertical Retrace Interrupt occurs and remains valid for the vertical period.

Bit 5 (Cursor Window Interrupt)
This bit is set whenever the cursor position (upper left corner of the cursor area) is moved into a new window.

Bit 4 (Y Break Interrupt)
This bit is set during the retrace period preceeding the display of a window that has its Interrupt Tag bit set.

Bit 3 (Last Break Processed Interrupt)
This bit is set whenever the VIEW retrieves a Y break with a terminator window number followed by an X break with a terminator window number. The purpose of this condition is to indicate that the VIEW will not access system memory again until the Start Delay count has expired. This can be used to give the system processor an opportunity to access the window list in memory. See section on Break Processing under Functional Description.

Bit 2 (Vertical Retrace Interrupt)
This bit is set when the vertical retrace interval begins.

INTERRUPT STATUS 2 (8 Bits—RR18[7,0])—Read only.

Bit 7 (Interrupt Pending)
This bit is set when any of the enabled interrupt causing conditions has occurred. This bit is reset by reading the Interrupt Status 1 register, a hardware Reset, or a software Reset command.

Bit 6-0 (Cursor Window Number)
These bits represent the window number that the VIEW cursor is currently resident in. This is determined with respect to the location of the upper left corner of the cursor block as specified in the Horizontal and Vertical Cursor Position registers. Bit 6 is the MSB and bit 0 is the LSB. If the cursor is resident in more than one window, the window number with the highest priority is reported here. The data in this register is valid from the time the Last Break Processed Interrupt occurs until the Start Delay count expires.

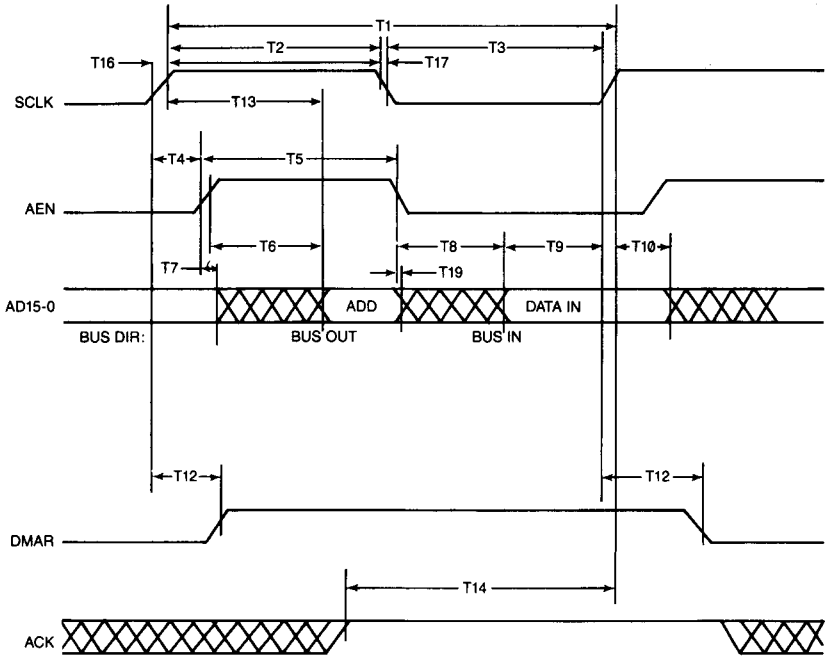


FIGURE 21—SYSTEM BUS MASTER TIMING

MAXIMUM GUARANTEED RATINGS*

Operating Temperature Range ..... 0°C to + 70°C
Storage Temperature Range ..... -55°C to + 150°C
Lead Temperature (Soldering, 10 sec) ..... + 300°C
Positive voltage on any pin (WRT ground) ..... Vcc + 0.3V
Negative voltage on any pin (WRT ground) ..... -0.3V
Maximum Vcc ..... + 7.0V

*Stresses above those listed may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operational sections of this specification is not implied.

NOTE: When powering this device from laboratory or system power supplies, it is important that the Absolute Maximum Ratings not be exceeded or device failure can result. Some power supplies exhibit voltage spikes or "glitches" on their outputs when the AC power is switched on and off. In addition, voltage transients on the AC power line may appear on the DC output. If this possibility exists it is suggested that a clamp circuit be used.

DC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5.0V \pm 5\%$

SYMBOL	PARAMETER	MIN	TYP	MAX	UNITS	COMMENTS
	Input voltage:					
V_{IL}	Low			0.8	V	All inputs except DCLK, SCLK, and $\overline{CS}$
V_{IH}	High	2.0			V	
V_{IL}	Low			1.0	V	DCLK, SCLK, $\overline{CS}$ with $V_{CC} = 5.0V$
V_{IH}	High	4.0			V	See NOTE 1.
	Output voltage:					
V_{OL}	Low			0.4	V	$I_{OL} = 1.6$ mA
V_{OH}	High	2.4			V	$I_{OH} = -40\ \mu A$
	Input leakage current:					
I_{IL}				10	μA	
I_{IH}				10	μA	
	Input/Output capacitance:					
C_{IN}				25	pF	All inputs
C_{OUT}				50	pF	All outputs
	Power supply current:					
I_{CC}				30	mA	

NOTES: 1. The V_{IH} MIN and V_{IL} MAX of SCLK, $\overline{CS}$, and DCLK are 80% and 20% of V_{CC} respectively.

AC ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5.0V \pm 5\%$

SYMBOL	PARAMETER	MIN	TYP	MAX	UNITS	COMMENTS
	System Bus:					
T1	SCLK period	200		10,000	ns	
T2	SCLK high	70		5,000	ns	
T3	SCLK low	70		5,000	ns	
T16	SCLK rise time			10	ns	
T17	SCLK fall time			10	ns	
T4	SCLK to AEN delay	0			ns	
T5	AEN pulse width	50			ns	
T6	AEN active to address valid delay			50	ns	
T7	AEN high to AD bus drive delay	0			ns	
T8	AEN low to AD bus float delay			50	ns	
T9	Data valid to SCLK setup time	35			ns	
T10	Data hold time from SCLK high	0			ns	
T12	SCLK to DMAR delay			65	ns	
T13	SCLK rising edge to address valid delay	50			ns	
T14	ACK to SCLK setup	35			ns	
T19	Address hold from AEN low	0			ns	
T18	$\overline{RESET}$ pulse width	200			ns	
T60	$\overline{CS}$ read pulse width	125			ns	
T61	$\overline{CS}$ write pulse width	75			ns	
T62	$\overline{CS}$ active to data valid delay			75	ns	
T63	$\overline{CS}$ active to AD bus drive delay	0			ns	
T64	$\overline{CS}$ inactive to AD bus float delay			75	ns	
T65	Write data setup time to $\overline{CS}$ inactive	40			ns	
T66	Write data hold time from $\overline{CS}$ inactive	0			ns	
T67	R/$\overline{W}$ and A/$\overline{D}$ to $\overline{CS}$ active setup time	30			ns	
T68	R/$\overline{W}$ and A/$\overline{D}$ to $\overline{CS}$ active hold time	0			ns	
T69	$\overline{CS}$ rise time			10	ns	
T70	$\overline{CS}$ fall time			10	ns	
T71	$\overline{CS}$ inactive between processor access	200			ns	
T72	Data hold time from $\overline{CS}$ inactive	0			ns	
	Display Bus:					
T30	DCLK period	100		5,000	ns	
T31	DCLK high	40		2,500	ns	
T32	DCLK low	40		2,500	ns	
T43	DCLK rise time			10	ns	
T44	DCLK fall time			10	ns	
T33	DCLK high to signal[1] valid delay			75	ns	
T34	Signal[1] hold time to DCLK rising edge	0			ns	
T45	Ext $\overline{VS}$/$\overline{HS}$ active to DCLK rising edge setup time	35			ns	
T46	DCLK high to Ext $\overline{VS}$/$\overline{HS}$ inactive hold time	0			ns	

NOTE: 1. Signal refers to following—ATTR7-0, CURS, HDA7-0, VDA11-0, CBLANK, $\overline{CYSN}$, RFRSH, BRKCHG, VLT, INT, $\overline{VS}$ and $\overline{HS}$ ($\overline{VS}$ and $\overline{HS}$ only when programmed as outputs).

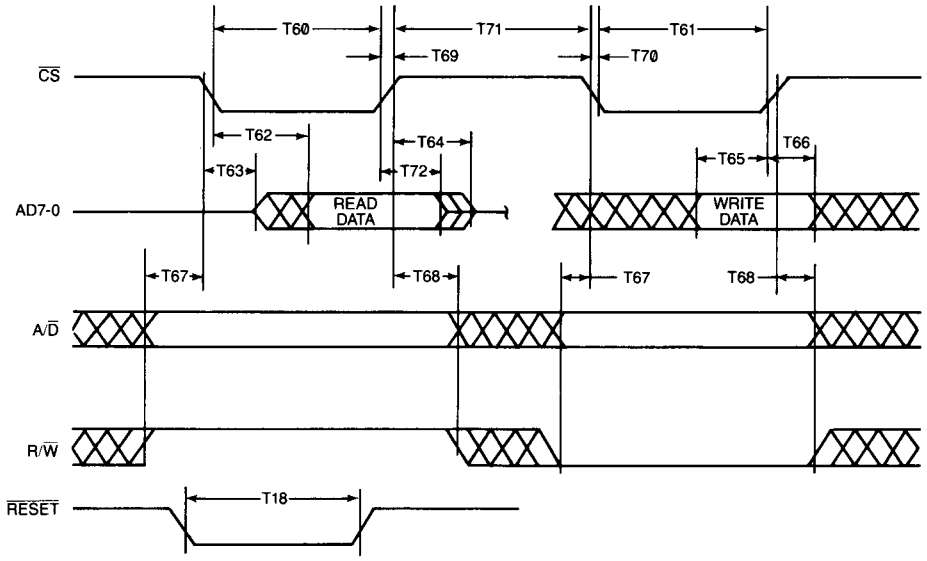
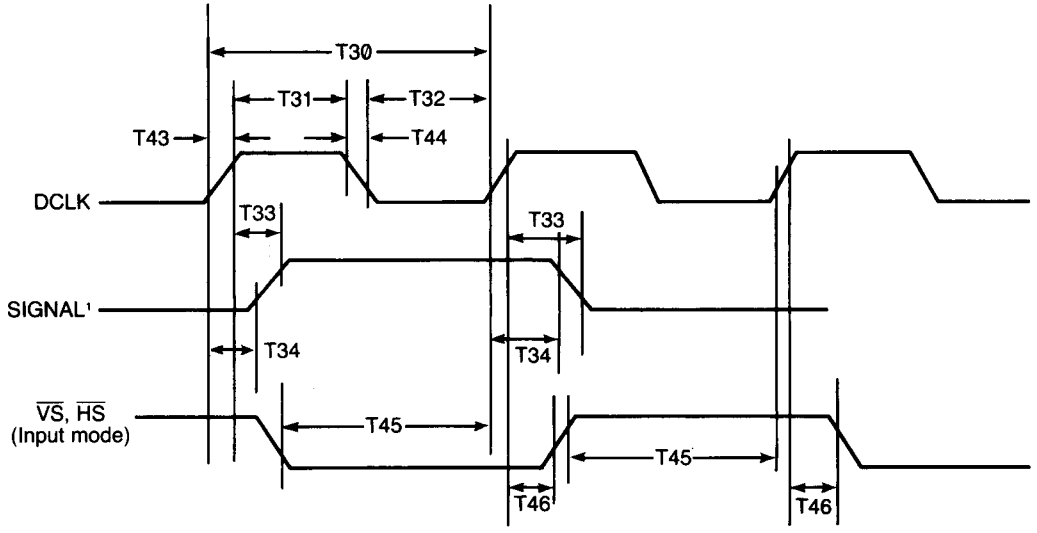


FIGURE 22—SYSTEM BUS PERIPHERAL TIMING



NOTE: ¹ SIGNAL above refers to ATTR7-0, CURS, HDA7-0, VDA11-0, CBLANK, CSYN, RFRSH, BRKCHG, VLT, INT, VS and HS (VS and HS only when programmed as outputs).

FIGURE 23—DISPLAY BUS TIMING