Zilog

Z8052 CRTC

Alphanumeric CRT Controller

PRELIMINARY PRODUCT SPECIFICATION

DISTINCTIVE CHARACTERISTICS

- On-chip DMA capability, operating via linked-list data structures
- Three on-chip row buffers, each 132 characters by 20 bits support split-screen soft-scrolling
- General purpose microprocessor interface. Compatible with 8086, Z8000® and 68000 CPUs.
- Soft-scrolling capability, with minimal CPU overhead
- Multiple vertical and horizontal screen divisions, with optional soft-scrolling within a window
- Character attributes (12 bits) can be invoked on a character by character basis
- Flexible vertical and horizontal sync control
- Flexible blanking for control of front and back porch positions
- Non-interlace, repeat field interlace, video interlace options
- High resolution five-bit character generator row addressing
- 16M byte system memory addressing capability
- Programmable blink options for cursors and characters

GENERAL DESCRIPTION

The Z8052 CRT Controller is a general purpose interface device for raster scan CRT displays. The CRTC provides efficient manipulation of complex character formats and screen structures to allow sophisticated text display without undue CPU overhead.

The CRTC is a register oriented product that is fully user programmable. The timing definition and operating modes are initialized by the host CPU. Display formats are real-time programmable on a row by row basis. Character attributes are specified on a character or field basis, and are interpreted and acted upon during active display of a character row.

Internal DMA capability assures efficient transfer of display information to the three on-chip line buffers. These three line buffers prevent screen flashing in split-screen soft-scrolling operations. The DMA loads the line buffers via linked list data blocks which facilitates editing and text composition.

The Z8052, in conjunction with the Z8152A bipolar video system controller, allows for the flexible assignment of visual attributes. The twelve attribute bits stored in the Z8052 include superscript, subscript, blink, highlight, reverse, underline, strike through and cursor. Both character and cursor can be made to blink at three different rates and the blink duty cycle is programmable. Further flexibility is achieved by the Z8152A which allows the video stream to be manipulated by selection of background and foreground as well as background/foreground reversal.

The Z8052 and Z8152A combination also supports proportional spacing, text justification and double width characters.

The Z8052 CRTC is assembled in a 68-pin LCC package, while the bipolar Z8152A VSC is assembled in a 48-pin DIP. These interface circuits are available as a chip-set for high performance CRT applications.

Figure 1. Z8052 Block Diagram

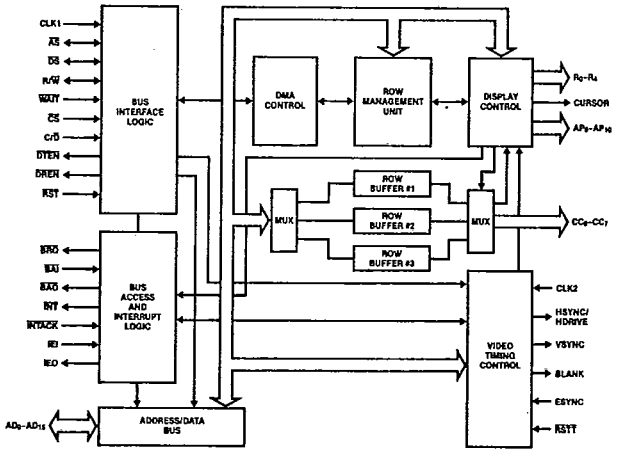
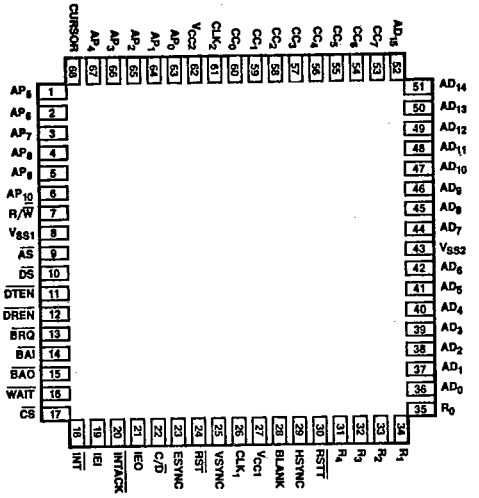
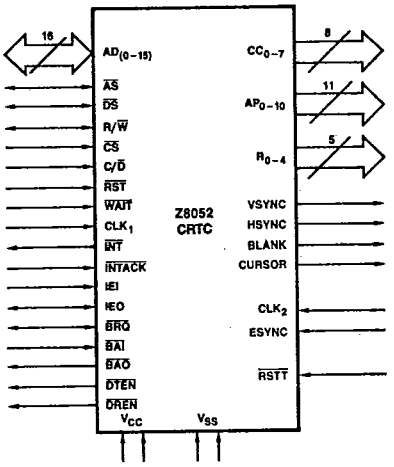


Figure 2. Z8052 Pin Assignments

BOTTOM VIEW



LOGIC SYMBOL



INTERFACE SIGNAL DESCRIPTION

All inputs and outputs of the CRTC are TTL compatible, unless otherwise indicated. Figure 2 shows the device pin-out.

V_{SS1}, V_{SS2} Ground

V_{CC1}, V_{CC2} +5V Power Supply

CLK_1 **Timing Clock, Input**
The Clock1 signal controls and times the DMA and peripheral portion of the CRTC. In proportional spacing applications, where CLK_2 is variable, CLK_1 must be used to time the horizontal and vertical sync rates. CLK_1 is non-TTL compatible, and is normally driven by the Z8152A/8153A VSC.

CLK_2 **Display Clock, Input**
The Clock2 signal is used to time character accesses from the CRTC line buffers. In applications which do not use proportional spacing CLK_2 is fixed in frequency and can be used to time horizontal and vertical sync rates, allowing CLK_1, the system clock, to be unrelated and asynchronous to the display timing. CLK_2 is non-TTL compatible, and should be driven by the VSC.

AD_0-AD_{15} **Address/Data Bus, Input/Output, Three-State**
The Address/Data Bus is a multiplexed, bidirectional, high-true, three-state bus. The presence of addresses is defined by the $\overline{AS}$ signal and the presence of data is defined by the $\overline{DS}$ signal. When the CRTC is in control of the system via its internal DMA capability, it controls the AD Bus; when the CRTC is idle, the CPU or other external devices control the AD Bus and may use it to access the internal registers of the CRTC. The high-order 8-bit memory address is output on the AD_0-AD_7 lines. Interrupt Vector information is also output on the AD_0-AD_7 lines.

$\overline{AS}$ **Address Strobe, Input/Output, Three-State**
Address strobe is a bidirectional, active-LOW, three-state signal. When the CRTC is in slave mode, and the bus master is accessing the CRTC's internal registers, $\overline{AS}$ can be used to optionally latch $\overline{CS}$ and C/$\overline{D}$ information during the first part of the transaction. During a DMA operation when the CRTC is in control of the system, $\overline{AS}$ is an output generated by the CRTC to indicate a valid address on the bus. In slave mode, the $\overline{AS}$ signal may be asynchronous to CLK_1.

$\overline{DS}$ **Data Strobe, Input/Output, Three-State**
Data Strobe is a bidirectional, active-LOW, three-state signal. When the CRTC is in slave mode, and the external system is transferring information to or from it, $\overline{DS}$ is a timing input used by the CRTC to move data to or from the AD bus. In the slave mode, the $\overline{DS}$ signal may be asynchronous to CLK_1. During a DMA operation when the CRTC is in control of the system, $\overline{DS}$ is an output generated by the CRTC and used by the system to move data onto the AD bus.

$\overline{CS}$ **Chip Select, Input**
The $\overline{CS}$ input is an active LOW signal used by the host processor to select the CRTC for a slave transfer.

$\overline{WAIT}$ **Wait, Input**
The $\overline{WAIT}$ input is an active LOW signal used to stretch the $\overline{DS}$ strobe whenever the CRTC has access to the host's bus for data transfer. The status of the $\overline{WAIT}$ signal is sampled on the falling edge CLK_1 during T_2 or T_W.

R/$\overline{W}$ **Read/$\overline{Write}$, Input/Output, Three-State**
Read/Write is a bidirectional, three-state signal indicating the data direction for the bus transaction under way, and remains stable for the length of the bus cycle. When $\overline{CS}$ input is active, Read (HIGH) indicates that the system is requesting data from the CRTC and Write (LOW) indicates that the system is presenting data to the CRTC. On the other hand, during a DMA operation when the CRTC is in control of the system, R/$\overline{W}$ is an output generated by the CRTC, with Read indicating that data is being requested by the CRTC from the addressed memory location, and Write indicating that the CRTC is driving a high-order address to an external latch.

$\overline{BRQ}$ **$\overline{Bus\ Request}$, Input/Output**
When the CRTC requires use of the bus for DMA activity, the $\overline{BRQ}$ line is driven LOW. It remains LOW until it has ceased using the bus.

$\overline{BAI}$ **$\overline{Bus\ Acknowledge\ In}$, Input**
Bus acknowledge In is an active LOW input. When the CRTC requires host bus access and has successfully pulled its $\overline{BRQ}$ pin LOW, a $\overline{BAI}$ LOW input signifies that the CRTC has obtained bus mastership after having internally synchronized its $\overline{BAI}$ active LOW input for two clock periods of CLK_1. The synchronization is required to alleviate metastable problems. When the CRTC does not require host bus access, the $\overline{BAI}$ input ripples to the $\overline{BAO}$. Forcing $\overline{BAI}$ HIGH would causes the Z8052 to relinquish the bus.

CURSOR **Cursor, Output**
This pin is the cursor output indicator.

ESYNC **External Sync, Input**
This pin is the external synchronization input line. If the ES bit in the mode register is set, the vertical frame scan will commence after the rising edge of ESYNC.

HSYNC **Horizontal Sync, Output**
HSYNC is an active HIGH output used to cause horizontal retrace of the CRT's electron beam. The output is held active LOW while the CRTC is reset to prevent unknown synchronization to the CRT which may cause damage to high bandwidth tubes. Note that this pin can also be initialized as Horizontal Drive.

VSYNC **Vertical Sync, Output**
VSYNC is an active HIGH output used to cause vertical retrace of the CRT's electron beam. VSYNC can be optionally synchronized by the ESYNC input. VSYNC is held LOW while the CRTC is reset to prevent damage to the CRT.

BLANK **Blank Video, Output**
BLANK is an active HIGH output. It serves to blank out inactive display areas of the CRT. The output is held active while the CRTC is reset.

R_0-R_4 **Row Control, Outputs**
R_0-R_4 outputs are active HIGH. These outputs represent the binary count of the active scan line being displayed. These outputs address the least significant address portion of an external character generator. The outputs are all held high for those scan lines that do not carry active video during normal character or superscript/ subscript display.

CC_0-CC_7 **Character Code Outputs**
CC_0-CC_7 outputs are active HIGH. The 8-bit character port, CC_0-CC_7, outputs eight bits of data stored in the character code section of the line buffer currently being displayed.

$\overline{INT}$ **Interrupt Request, Output, Open Drain**
This line is used to indicate an interrupt request to the host processor. It is driven LOW by the CRTC, until an interrupt acknowledge is received on the $\overline{INTACK}$ pin, or the relevant IP or IE bits in Mode Register 2 are reset.

$\overline{INTACK}$ **Interrupt Acknowledge, Input**
When $\overline{INTACK}$ is driven LOW, the CRTC examines its IEI line to determine whether it has been granted an acknowledge by the CPU. It also starts priority resolution of the daisy chain. When $\overline{DS}$ is active the vector is placed on the bus if enabled.

IEI **Interrupt Enable-In, Input**
A HIGH on IEI during an Interrupt Acknowledge cycle is regarded as an interrupt acknowledge to the CRTC. A LOW on IEI during Interrupt Acknowledge signifies that a higher priority interrupt on the daisy chain is being acknowledged.

IEO **Interrupt Enable-Out, Output**
IEO follows IEI during Interrupt Acknowledge if the CRTC has not made an interrupt request. IEO LOW disables lower priority devices from making interrupt requests.

$\overline{DTEN}$, $\overline{DREN}$ **Data Transmit Enable, Data Receive Enable, Outputs, Open Drain**
Data Transmit Enable and Data Receive Enable are used to control bus transceivers external to the CRTC, should they be required. When $\overline{DTEN}$ is low, the transceivers should transmit from the CRTC onto the bus. When $\overline{DREN}$ is low, the transceivers should receive data from the bus. $\overline{DTEN}$ and $\overline{DREN}$ are never low simultaneously.

C/$\overline{D}$ **Command/$\overline{Data}$, Input**
C/$\overline{D}$ is used by the CRTC when in slave mode, to determine if an I/O transaction with the host CPU is transferring a command or data. When the CRTC is not involved in an I/O transaction with the host, C/$\overline{D}$ is disregarded.

AP_0-AP_{10} **Attribute Port Outputs**
These 11 lines are used to display character attribute information synchronous with each character and CLK_2. During horizontal SYNC, the row attribute information contained in the Row Redefinition Block is output on AP_0—AP_{10}.

$\overline{BAO}$ **$\overline{Bus\ Acknowledge\ Out}$, Output**
$\overline{BAO}$ output is forced active HIGH when the CRTC requests bus mastership, otherwise the $\overline{BAI}$ input ripples out of the CRTC via the $\overline{BAO}$ output.

$\overline{RST}$ **$\overline{Reset}$, Input**
A Low on this input for at least 5 clock cycles is interpreted as a reset signal. The effect of reset is to drive all CRTC bus signals into the high-impedance state, to clear all mode bits except bits 9 through 15 in MR2, and to force the CRTC into slave mode.

$\overline{RSTT}$ **$\overline{Test\ Reset}$, Input**
For test use only. This pin is a No Connect.

TABLE 1.

Attribute	Effect
Reverse	– Causes the designated character to be displayed in reverse video.
Highlight	– Highlights the applicable character.
Blink	– Blinks the designated character at one of four programmed blink rates.
Underline	– Underlines the designated character at a programmable scan line.
Subscript	– Causes the character to be displayed as a subscript.
Superscript	– Causes the character to be displayed as a superscript.
Shifted Underline	– A second underline.
Cursor	– Causes the attribute or X-Y cursor to be displayed at the designated character position.
Latched	– Indicates that the attribute should be latched for all successive characters until changed.
Ignore	– Causes the CRTC to skip over the designated characters. Useful for embedded control characters and protected fields that do not get displayed.
User Definable	– Four attribute bits reserved for user definition.

FUNCTIONAL OVERVIEW

The block diagram of the Z8052 CRTC is shown in Figure 1. Communication with the external host system takes place over the 16-bit Address/Data bus, $\overline{AD}_0-\overline{AD}_{15}$. Transfers over the AD bus are controlled by the $\overline{CS}$, C/$\overline{D}$, $\overline{AS}$, $\overline{DS}$, and R/$\overline{W}$ lines. When the CRTC is in slave mode, these four bus control lines are inputs. When the CRTC is in DMA mode. $\overline{AS}$, R/$\overline{W}$ and $\overline{DS}$ are outputs and control the external bus.

Following reset, the host system initializes the CRTC's timing and control registers, as well as one address pointer to the start of the display data location in the host memory. Following initialization and upon command from the host, the CRTC takes over bus control from the host and transfers display row control data, character code, and character attribute data. The CRTC requests the host bus by sampling the $\overline{BRQ}$ line for activity; if the $\overline{BRQ}$ line is HIGH, the CRTC drives it LOW, and also drives $\overline{BAO}$ HIGH, to obtain priority over lower priority bus requestors. The on-chip DMA Controller circuit controls the data transfer and performs character data loading into the on-board line buffers.

The CRTC is real-time programmable on a character row by row basis through a row control data block fetched either from the host memory or from a dedicated display memory. The row control block contains address links to the next row's row control block, a character and attribute data address for the current row and other pertinent control functions for the row. Data from the row control block is transfered into the appropriate set of registers for active control of display and data fetch operations during the subsequent display of character row data. A Top Of Page register contains the address of the Main Definition Block for the screen. The Main Definition Block in turn, points to the first Row Control Block. The character row data, comprised of character code and attribute (if the latter is specified), is fetched starting at the address and for the character length obtained from the Row Control Block. The character code and its attribute consist of a 20-bit wide word which is stored, FIFO style, into one of the three on-board 132-character by 20-bit line buffers. Character attributes are on a character by character basis and are interpreted and acted upon by the CRTC during the active display period of the contents of a line buffer. Output lines CC_0-CC_7 form the transfer path for character code data to an external matrix type character generator, while the character attribute, after selective masking, is interpreted and combined with the resulting video.

Output lines R_0-R_4 exhibit the scan line number for the specific character being displayed, while the character Row Control logic allows alteration of the scan line number output at the R_0-R_4 lines to enable the display of normal superscript or subscript characters.

The HSYNC, VSYNC and BLANK output lines provide the CRT synchronization signals. The Horizontal and Vertical Control logic blocks contain counters and host programmable registers for deriving the timing signals from either the CLK_1 or the CLK_2 input as well as an ESYNC input line for frame synchronization to an external source, such as the power line frequency. CLK_2 runs at the display character rate. It is a submultiple of the dot clock, whose frequency is determined by the Z8152A oscillator. CLK_2 controls the CRT synchronization lines HSYNC and VSYNC, as well as BLANK, and the rate of character output from the CRTC. CLK_1, which may be asynchronous to CLK_2, controls all DMA and related bus activity, associated with the CRTC. In proportional spacing applications CLK_1 may be also used to time the synchronization signals.

CHARACTER ATTRIBUTES

Character attributes affect various CRTC output signals and other operations on a character by character basis. Each attribute word occupies a 16-bit word in memory. Each character, however, need not invoke a new attribute.

Character attributes are stored in parallel with the corresponding character code in each line buffer.

The character attribute information which makes up the character attribute word is shown below:

AW_{15}	Latched/unlatched	AW_7	User definable
AW_{14}	Cursor/definable	AW_6	Highlight
AW_{13}	Ignore	AW_5	Reverse
AW_{12}	Reserved	AW_4	Superscript
AW_{11}	Reserved	AW_3	Subscript
AW_{10}	User definable	AW_2	Shifted underline/ strike through
AW_9	User definable	AW_1	Underline
AW_8	User definable	AW_0	Blink

DESCRIPTION OF CHARACTER ATTRIBUTES

LATCHED/$\overline{\text{UNLATCHED}}$

When this bit is set to 1 ("latched") the attribute information applies to all characters following the character that invoked the attribute word. Only the presence of a further latched attribute word cancels the effect of a previous latched attribute word. If the Latched/$\overline{\text{Unlatched}}$ bit is set to 0, ("unlatched")

then the attribute information only applies to the character that invoked the attribute word. All successive characters are modified by the latched attribute information that was valid prior to the unlatched attribute word. The Latched/Unlatched bit is not output to the attribute port. The initial state of the latched attribute value is undefined. At the start of any horizontal line, the latched attribute information is the same as at the end of the previous line, unless changed by a further latched attribute.

CURSOR

If this bit is set, then a cursor is displayed at the affected character position(s), dependent upon the mode of the cursor display logic. See the section on cursor display for further details.

IGNORE

When the ignore is set, it inhibits the loading of the associated character into the CRTC line buffer. Such character(s) may be used as control character or software tags, and are not displayed. Whenever the ignore encoding is detected, both the attribute word and its associated character code are not written into the line buffer, unless the DH (Display Hidden) bit in Mode Register 1 is set. Note that the ignore bit is not brought out to the attribute port.

USER DEFINABLE

The AW_7-AW_{10} attribute bits provide 4 bits of user definable attribute information. These bits are directly output on pins AP_7-AP_{10} of the attribute port. (In addition to these four user definable attribute bits, the cursor bit can also be user definable under certain conditions.)

HIGHLIGHT

When this bit is set and AP_6 is connected to the FS input of the Z8152A the character is displayed highlighted. The AP_6 pin of the attribute port goes active for each scan line of the relevant character(s).

REVERSE

When this bit is set and AP_5 is connected to the REV input of the Z8152A, the character is displayed reversed. The AP_5 pin of the attribute port goes active for each scan line of the relevant character(s).

SUPERSCRIPT

When this bit is set to 1, the affected character is displayed as a superscript. Its position on the character row (R_0-R_4) is determined by the superscript control field in the row redefinition block, for that particular row.

SUBSCRIPT

When this bit is set, the affected character is displayed as a subscript. Its position on the character row (R_0-R_4) is determined by the subscript control fields in the row redefinition block.

UNDERLINE/SHIFTED UNDERLINE

Attribute bits AW_1 and AW_2 provide underline and shifted underline display. The underline/shifted underline display information is output on the AP_1 and AP_2 attribute port pins, during applicable scan lines of the character. (The applicable scan lines have been programmed within the row redefinition blocks.)

BLINK

When this attribute is invoked, the attribute port pin AP_0 is gated with the character blink rate generator, during the time that the relevant character is output on CC_0-CC_7.

The character blink rate and character blink duty cycle are derived from the blink field of the Main Definition block.

ATTRIBUTE FETCHES

Attributes can be fetched in three different ways to suit most design philosophies (see Figure 4). In Option 1, one attribute is fetched per character. This option, although straightforward, imposes heavy bus overhead since the DMA has to access the attribute list from memory for every character displayed on the screen. Bus overhead can be reduced considerably by fetching attributes on a demand basis. Option 2 and 3 accomplish this in two different ways. In Option 2 one character bit is set to 1 when an attribute is required. When this bit is set to 0, the attribute will not be fetched. This option allows 7 bits of character code or a 128 character set for display with no overhead for attribute incorporation.

Option 3 makes use of an 8-bit flag which precedes the character invoking the attribute. This option allows for a 255 character set with an 8-bit overhead (the flag) per attribute.

CURSOR GENERATION

The CRTC can generate three different cursor formats: block, underline, and reverse, at variable blink rates and blink duty cycles.

Cursor information for the CRTC comes from two different sources, and each source can be independently steered to one of three different destinations. The two cursor sources are:

1. The XY cursor field which is held in the Main Definition Block for the screen.
2. Attribute Word bit 14 of the character attribute word. A cursor designated by an attribute will follow its row and character position whenever text is scrolled. The cursor controlled by positioning X and Y coordinates within the cursor X and Y register will be displayed on a fixed X, Y character position on the screen. The X, Y cursor should be disabled by resetting the CUE bit in Mode Register 2 during soft scroll.

Figure 3. Z8052 Attribute Word

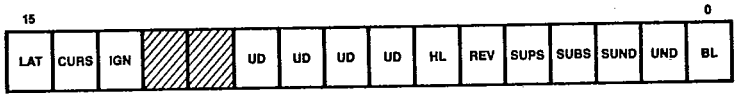
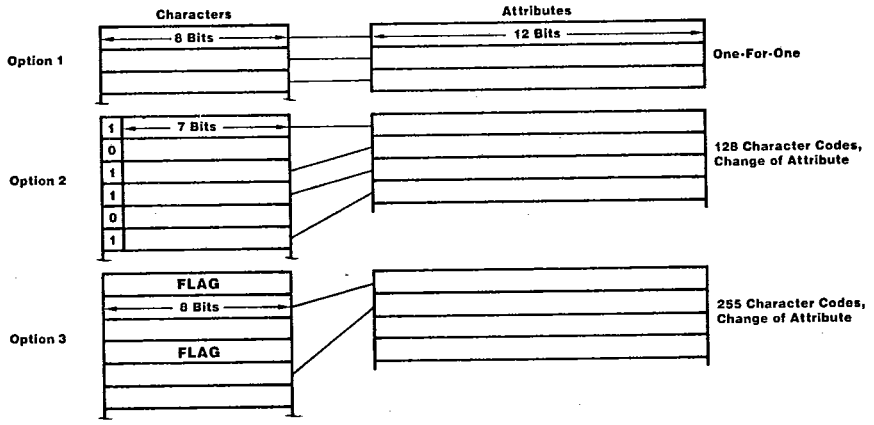


Figure 4. Attribute Fetch



The steering of the cursor sources is under software control of the cursor mask field within mode register 2. The field is divided into two three-bit segments, one for the XY cursor and one for the attribute cursor. Three destinations are selectable for each cursor source:

(a) The cursor pin
(b) The underline pin
(c) The reverse video pin.

If (a) is selected, then either the whole character cell or partial character cell is selectable. If whole is selected, the cursor pin will be active for every scan line of the character cell. If part is selected, then the cursor pin will only be active for those scan lines within the limits of CURSOR START and CURSOR END, as specified in the row-redefinition information.

If (b) is selected, then either an underline will be active, if CURSOR START and END have the same values, or a block, if CURSOR START and END are not coincident.

If (c) is selected, then either all or part of the character will be reversed, dependent upon the CURSOR START and CURSOR END setting as explained in (b).

In addition to these choices, either cursor can be made to blink, at the cursor blink rate, and duty cycle, as programmed into the main definition block blink field.

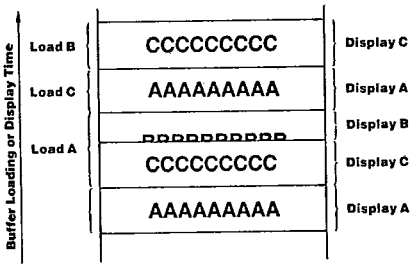
ROW BUFFERS

The on-chip DMA controller accesses the display memory and loads data from linked-list data blocks in memory into one of three row buffers. Each line buffer is 132 characters in length and 20 bits wide. Each 20-bit wide location accommodates an 8-bit character code and 12-bit attribute words. The row buffers operate in a rotating fill-display mode whereby one buffer is being loaded while another is being displayed.

The presence of three row buffers on-chip is of significant advantage in split screen soft-scrolling operations where a character row may only be displayed for a single scan line. With two row buffers, this would not leave enough time for the reloading of the alternate line buffer. A partially filled buffer results in screen flashing. This can only be prevented by incorporating three line buffers. Figure 5 highlights this advantage.

Note that only the shaded row buffers are scrolling on the horizontally split screen.

Figure 5. Triple Row Buffers



In the rotating fill-display mode, Row Buffer C is displayed when Row Buffer B is being loaded. Likewise the next Row Buffer C is loaded while Row Buffer A is being displayed. Because of the split-screen, Row Buffer B is displayed for one scan line only, while Row Buffer A is being loaded. By virtue of the third row buffer the loading of Buffer A can spill over into the next buffer display, thus eliminating screen flashing.

SOFT-SCROLLING

A soft-scroll is defined as the gradual displacement of a character row on a scan line by scan line basis. Soft-scrolling is achieved by a gradual offsetting of the scan line counter, on a frame by frame basis. At the start of the scroll, the offset counter is set to zero or equal to the number of scan lines per character row depending on whether the scroll is up or down. As the counter is incremented or decremented, the text travels up or down until the offset is equal to the number of scan lines or zero. The start of the screen pointer pointing to the character row is adjusted and the offset counter reset simultaneously to scroll the next successive character row. Soft-scrolling of the entire screen is thus a simple task. However, implementing a split screen soft-scroll is cost prohibitive and complex in MSI.

A number of applications require screen overlays, such as menu or status areas which must remain static while the major portion of the screen is scrolling or vice versa. The Z8052 can support multiple windows, with each being independently scrollable. This feature is described in detail in the following section.

LINKED LIST DATA STRUCTURES

The DMA channel on the Z8052 operates via linked list structures that allow for the overlaying and independent soft-scrolling of windows. The linked list data structures are particularly suited to the manipulation of data strings where insertions and deletions are common. A typical CRTC Linked List Structure is shown in Figure 6.

The linked list consists of Row Control Blocks (RCBs) for each character row on the screen. The RCB does not contain any displayable data, but contains the address which points to the character information. Each RCB is linked to the next block via an address link word (RCB ADR). The structure of the RCB linkage is shown in Figure 7. The Top of Page register on-chip points to the Main Definition Block which in turn points to a linked list of RCBs.

The Z8052 allows for the separation of attribute and character lists. By extending the RCB, split screen segments can be constructed as in the case of RCB_2 in Figure 7. In parallel with the screen or background data structure, there exists a window structure which contains Window Control Blocks (WCBs) for each row of each window. Windows can exist in any position on the screen and are overlayed on top of the screen or background information. For example, the structure shown in Figure 8 could be used to implement a menu overlay at the top of the screen together with a status overlay.

MAIN DEFINITION BLOCK

The Main Definition Block is a set of control data and addresses, located in the system memory, which allow the user to specify screen oriented features. The TOP OF PAGE register points to the first word of the Main Definition Block. Cursor position, fill code and scroll rate are set by the appropriate fields within the block. The Main Definition Block also points to the first Row Control Block.

Figure 7. Background Data Structure

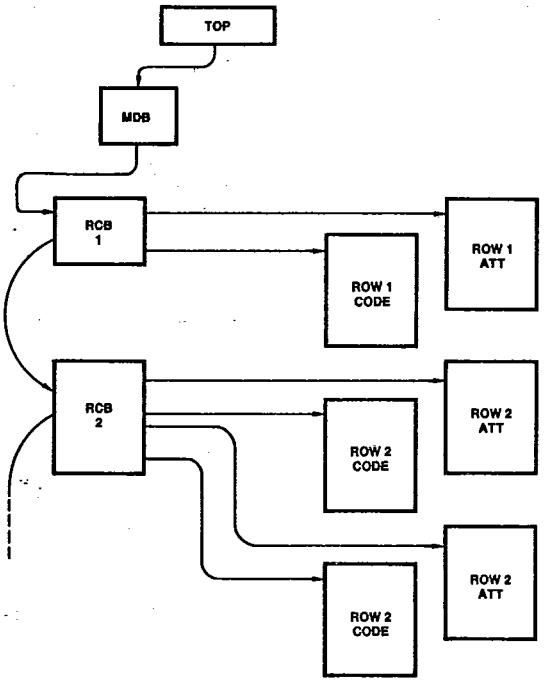


Figure 6. Z8052 Linked List Structure

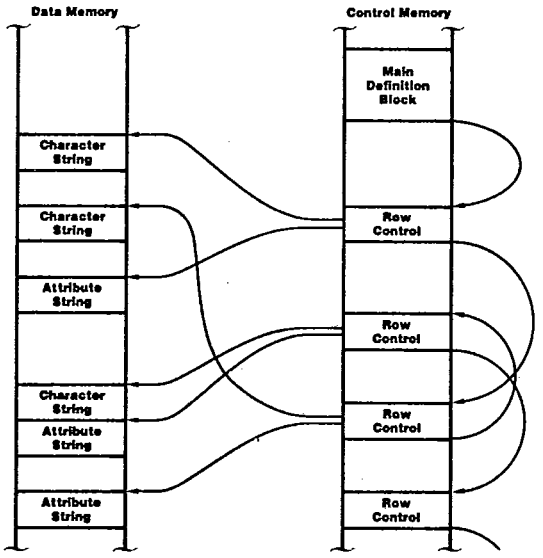
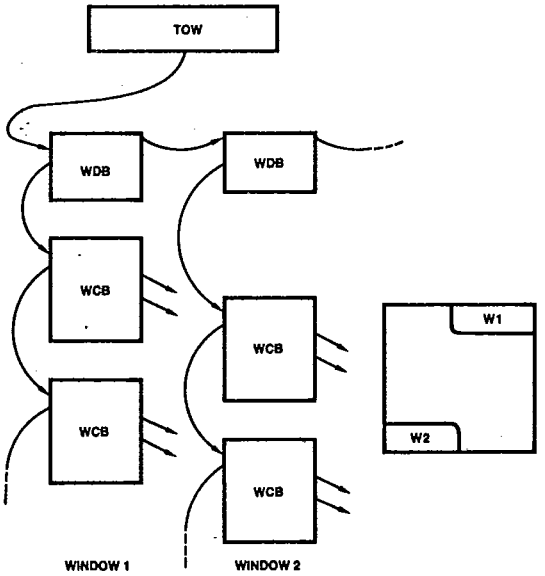


Figure 8. Window Data Structure



ROW CONTROL BLOCKS

The RCB ADR POINTER in the Main Definition Block points to the first word of the first Row Control Block of the list. Each Row Control Block in the main chain is linked to the next via the RCB ADR (Row Control Block Address) word address pointer. Changing the RCB Link Pointer within a Main Control Block allows quick insertion or deletion of a character row (line).

Attributes associated with character data exist in their own separate lists. A character row may be composed of one or more segments of data. Each segment is a block of words with consecutive addresses. A Row Control Block has a character code and character attribute segment start address pointer: SEG CODE ADR for character codes (2 character codes/word) and SEG ATR ADR for character attributes. A third word, SEG LENGTH and SEG DISPLAY defines the number of characters (byte count) contained in the segment as well as the number of displayed characters in the segment. Character attributes are in word format and there can be as many character attributes as character codes.

WINDOW DEFINITION BLOCK

The Window Definition Block defines the size and location of the window. It is the header block to a list of Window Row Control Blocks and can also point to another Window Definition Block if more than one window is displayed on the screen. The TOP OF WINDOW register points to the first word of the first Window Definition Block. Within the first Window Definition Block, the WCB ADR points to the current window's first Window Row Control Block, while the NEXT WDB ADR points to the next window's Window Definition Block. Window size is specified by two words in the Window Definition Block. STRT WINDOW ROW # and END WINDOW ROW # are byte values which position the window vertically on the screen. The window display becomes active in the character row number specified by STRT WINDOW ROW #, and will become inactive in the character row following END WINDOW ROW #.

WINDOW ROW CONTROL BLOCKS

The Window Row Control Blocks have the same format as the Row Control Blocks.

The WCB Link Pointer is the address link to the next row's Window Row Control Block. A window can also be described with segments, and the Window Row Control block contains several words for each segment: WSEG CODE ADR, WSEG ATR ADR, and WSEG LENGTH.

To hard-scroll a window, it is only necessary to change the WCB ADR in the Window Definition Block to a different Window Row Control Block.

WINDOW DISPLAY MECHANISM

A window is any bounded area on the screen which is linked in by a Window Definition Block. The window has the following size characteristics:

- Width: Defined by the number of character code positions occupied within a character row. Maximum width is the length of the line buffer (132 characters), and minimum width is one character.
- Height: Defined by the number of displayable character rows contained within the window. The maximum height is the total number of displayed character rows on the face of the screen. The height limit is specified by the number of Window Row Control Blocks in the window linked list. The minimum height of a window is one row.

Window Positioning

The window is originally positioned to occupy any portion of the displayable character rows. It can be as large as the full screen or as small as one row high and one character wide. The window is always unscrolled when first displayed. (The counter holding the value of the first scan line of the uppermost character row of the window is reset.)

The window must be positioned horizontally such that its left- and right-hand sides begin and end at a main character row segment boundary. Any unfilled character positions within the window segment, and following the end of the window segment to the end of the line buffer (character position 131) are filled with the fill character code, obtained from the Main Definition Block.

Multiple Windows

Multiple windows can be displayed simultaneously, but only one window can be soft-scrolled at any particular time. Windows cannot be horizontally aligned to each other, and hence must be specified on non-overlapping character row boundaries (see section on virtual windows). Each window is defined by a Window Definition Block, and the scrolling window is designated by a control bit within the Window Definition Block.

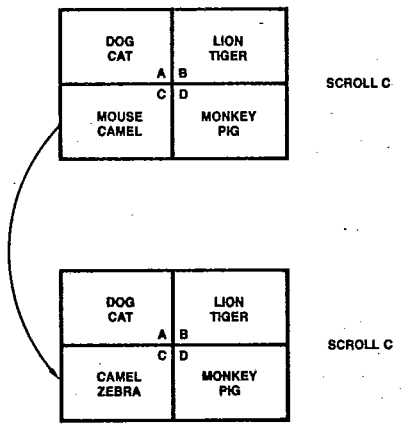
WINDOW POSITIONING

The window position is defined in the Window Definition Block. The coordinate units are background character rows (Y START), (Y END), and background character columns (X START), (X END). When the background is scrolling the window (or windows) should remain stationary on the display.

EXAMPLE OF WINDOW OVERLAYS

The example below (Figure 9) explains how windows are constructed using the Linked List feature that the Z8052 provides.

Figure 9. Example of Vertical Split Screen Smooth Scroll



Step 1

The first step toward constructing windows on a CRT screen is to split the screen horizontally and vertically using row control blocks with multiple data pointers. The data pointers in each RCB point to the first characters within each subscreen area defined by the horizontal/vertical splits. In this example, the RCB that controls the first character row (DOG/LION) contains two data pointers. The first points to subscreen DOG and the second to subscreen LION. The SEG LENGTH information in the RCB indicates to the DMA to switch from data field DOG to data field LION. The Linked List Structure for this example is shown in Figure 10. Note that in most applications, this split screen will have been set up prior to the invocation of the window.

Step 2

A window can now be overlayed on to the background by the creation of a window linked-list as shown in Figure 11. The scrollable window has a linked list structure pointed to by the Top of Window (TOW) pointer which functions similarly to TOP. The other information required for window definition is the STRT WINDOW CHAR # and END WINDOW CHAR # which define the start/end coordinates of the window. To effect a window scroll just one change to the TOW value is required which significantly relieves CPU overhead.

Virtual Windows

Although the rules of multiple windows do not permit overlapping windows, the background and window structures can be used to implement virtual horizontally aligned windows. This can be best described by using the illustration in Figure 9. The screen is divided into 4 subscreens A, B, C and D, each of which can be independently defined as windows using the Linked List Structures similar to Figure 11.

If subscreen C is defined as a window, subscreen A, B and D can be configured to be the background. Window C can be scrolled independently of the background by TOW pointer manipulation. Similarly subscreen D can be defined as a window with A, B and C configured as background. Thus, two aligned subscreens can be independently defined as windows by intelligent use of linked list structures, giving the user the illusion of aligned windows.

Figure 10. Split Screen Control Blocks

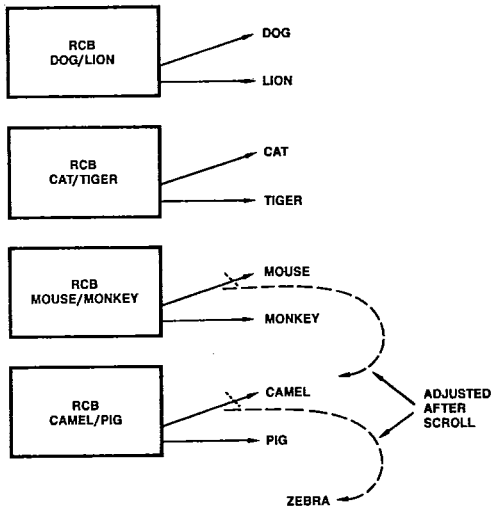
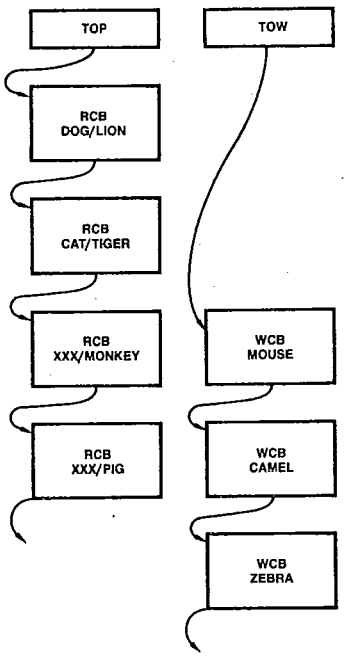


Figure 11. Window Overlay Structure



HORIZONTAL SCREEN FORMAT

The horizontal format defines the general timing of a single raster scan line. The scan line consists of two basic periods: visible raster line scan from left-to-right across the CRT screen and the right-to-left beam retrace period (or horizontal sync). The beam is always blanked during the retrace period. The front and back porch periods on either side of the horizontal sync are also blanked because no active video is desired during that time.

Horizontal scan frequencies range from a minimum of 15kHz for small screen, low bandwidth CRTs up to about 60kHz for 100MHz bandwidth large screen CRTs. The horizontal format versatility must accommodate this wide range scan frequencies. The horizontal circuit generates three basic timing signals: horizontal sync, horizontal drive, and blanking. Either horizontal sync or horizontal drive is output at the HSYNC pin as selected by a bit in the mode register. The horizontal blanking signal is "ored" with the vertical blanking signal prior to output at the BLANK pin.

HORIZONTAL TIMING CONTROL

Horizontal timing is controlled by the $\overline{RST}$ signal and the DE (Display Enable) bit in the mode register.

The HSYNC output is disabled (inactive) and the BLANK output active whenever the CRTC is reset by $\overline{RST}$ input (active low) or whenever the DE bit is reset (display disabled). $\overline{RST}$ active low is a hardware reset to the CRTC (this action also resets DE bit), and the DE bit is a software reset of the CRTC.

Z8052 VERTICAL SCREEN FORMAT

The vertical format defines the number of horizontal scan lines to be displayed in each frame. The front and rear porches, as well as the vertical retrace time, are also defined.

The CRTC operates in either an interlace or non-interlace mode. The I_1 bit in Mode Register 1 determines if the CRTC will operate in the interlace or non-interlace mode. See below for each of the interlace options.

The Vertical Line Counter is clocked by either the horizontal sync rate in the non-interlaced or twice the horizontal sync rate in the interlaced mode. In non-interlaced mode all vertical frames (period between two vertical sync pulses) are *even*. In interlaced mode, the first vertical frame following a Display Enable (setting of DE bit in the Mode register) is always *even* and alternates between odd and even from there on.

EXTERNAL SYNC OPERATION

The ESYNC input allows synchronization of the CRT display vertical frame rate to the power line frequency to eliminate interference effects. The ES bit in Mode register specifies whether the ESYNC input is used to control the Vertical Sync rate.

The ESYNC input is recognized by the CRTC during every frame. It causes the VSYNC signal to become active at the occurrence of HSYNC. In non-interlaced mode, VSYNC becomes active at the rising edge of HSYNC active; in interlaced mode, VSYNC becomes active at the next HSYNC, active when in the even frame, or the next half point between HSYNCs (2x HSYNC) in the odd frame.

INTERLACE

There are two types of interlace, Repeat Field Interlace (RFI) and Interlaced Video (IV). The effect of both schemes is to offset the vertical position of the scan lines of the odd numbered fields so that they will be physically interleaved with the scan lines of the even fields. For RFI, the same video information is displayed on both odd and even fields. The slight offset of the odd field tending to eliminate the horizontal stripes that sometimes occur between scan lines of non-interlaced displays.

Interlaced Video is used to increase the amount of information displayed on a monitor without increasing the horizontal or vertical scan rates. IV takes advantage of the Odd field scan line offset by displaying half the video in the even field (alternating lines) and half in the odd field. The effect is to essentially double the vertical character density with respect to RFI or non-interlace.

REGISTER SUMMARY

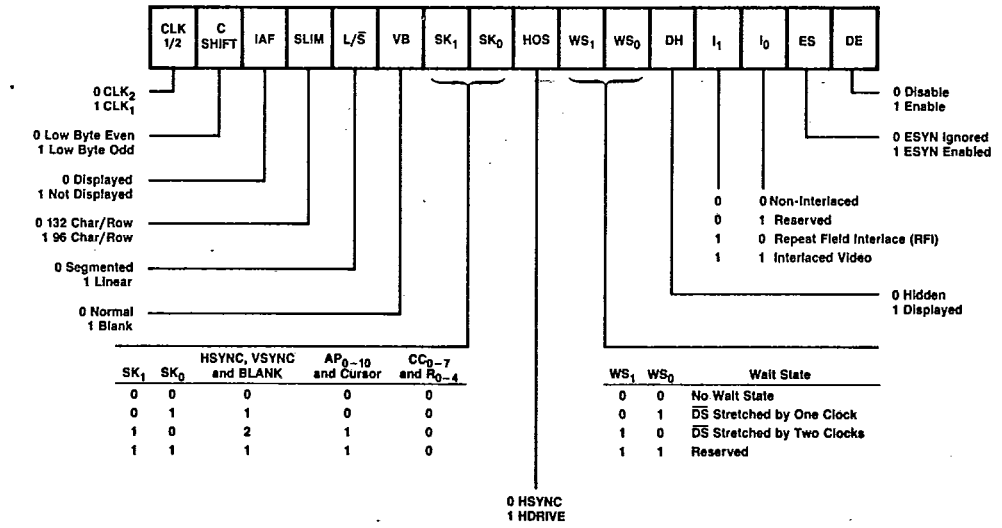
ADDRESS ($AD_4 - AD_0$)

BINARY	HEX	TYPE	ACTIVE BITS	REGISTER MODE
0 0 0 0 0	00	R/W	16	MODE 1
0 0 0 0 1	01	R/W	16	MODE 2
0 0 0 1 0	02	W	12	ATTRIBUTE ENABLE
0 0 0 1 1	03	W	5	ATTRIBUTE REDEFINITION
0 0 1 0 0	04	R/W	8	TOP OF PAGE SOFT (HI-ORDER)
0 0 1 0 1	05	R/W	16	TOP OF PAGE SOFT (LO-ORDER)
0 0 1 1 0	06	R/W	8	TOP OF WINDOW SOFT (HI-ORDER)
0 0 1 1 1	07	R/W	16	TOP OF WINDOW SOFT (LO-ORDER)
0 1 0 0 0	08	W	16	ATTRIBUTE FLAG
0 1 0 0 1	09	R/W	8	TOP OF PAGE HARD (HI)
0 1 0 1 0	0A	R/W	16	TOP OF PAGE HARD (LO)
0 1 0 1 1	0B	R/W	8	TOP OF WINDOW HARD (HI)
0 1 1 0 0	0C	R/W	16	TOP OF WINDOW HARD (LO)
1 0 0 0 0	10	W	16	DMA BURST
1 0 0 0 1	11	W	12	*VSYNC WIDTH/SCAN DELAY
1 0 0 1 0	12	W	12	*VERTICAL ACTIVE LINES
1 0 0 1 1	13	W	12	*VERTICAL TOTAL LINES
1 0 1 0 0	14	W	16	*HSYNC/VERTINT
1 0 1 0 1	15	W	9	*HDRIVE
1 0 1 1 0	16	W	9	*H SCAN DELAY
1 0 1 1 1	17	W	10	*H TOTAL COUNT
1 1 0 0 0	18	W	10	*H TOTAL DISPLAY

*These registers should only be accessed when Display Enable ("DE" bit in Mode Register 1) is reset, since they control the video timing signals.

MODE REGISTER 1

ADDRESS: 0 0 0 0
READ/WRITE

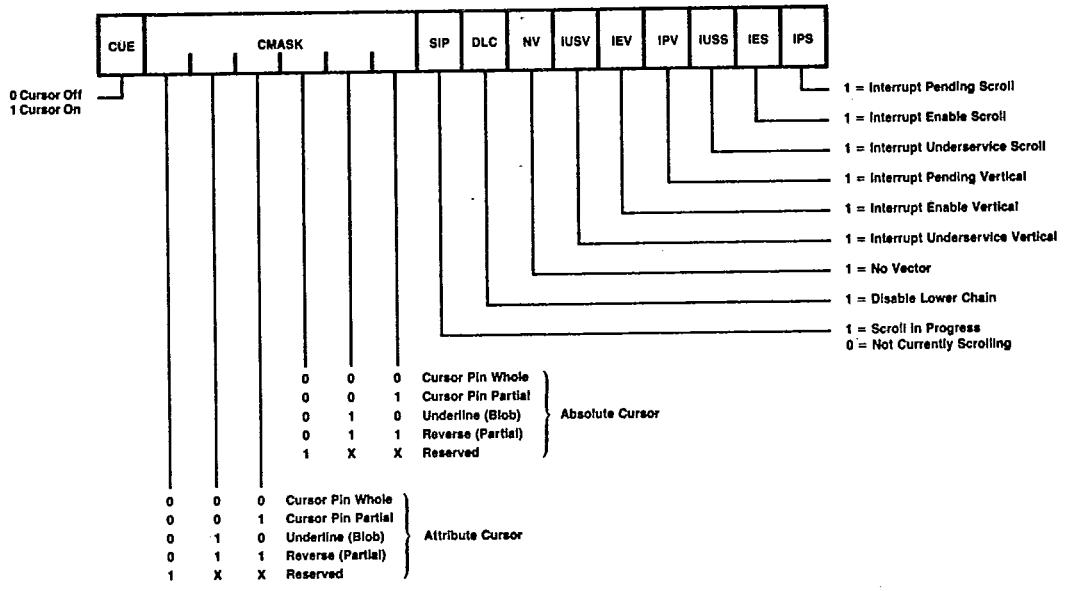


SK_1	SK_0	HSYNC, VSYNC and BLANK	AP_{0-10} and Cursor	CC_{0-7} and R_{0-4}
0	0	0	0	0
0	1	1	0	0
1	0	2	1	0
1	1	1	1	0

WS_1	WS_0	Wait State
0	0	No Wait State
0	1	$\overline{DS}$ Stretched by One Clock
1	0	$\overline{DS}$ Stretched by Two Clocks
1	1	Reserved

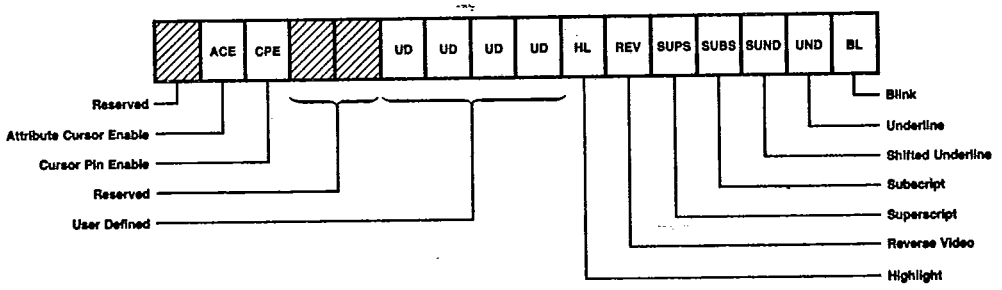
MODE REGISTER 2

ADDRESS: 0 0 0 0 1
READ/WRITE



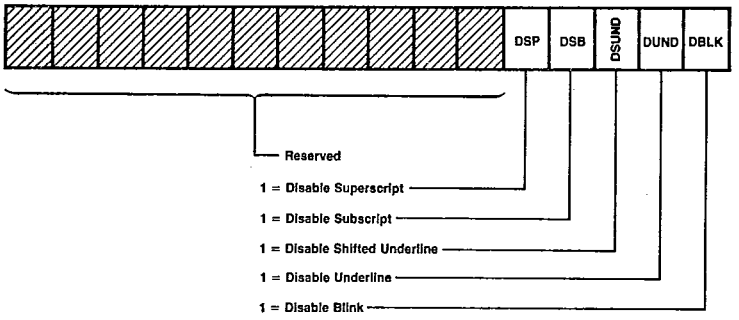
ATTRIBUTE PORT ENABLE REGISTER

ADDRESS: 0 0 0 1 0
WRITE



ATTRIBUTE REDEFINITION REGISTER

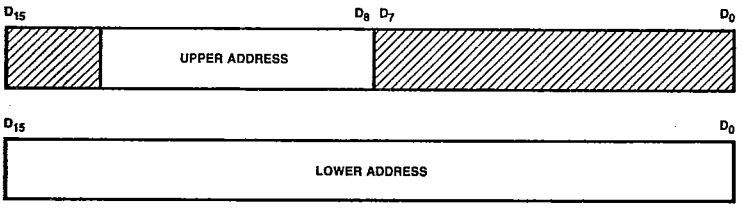
ADDRESS: 0 0 0 1 1
WRITE



TOP OF PAGE/TOP OF WINDOW REGISTERS L/$\overline{S}$ = 0

READ/WRITE

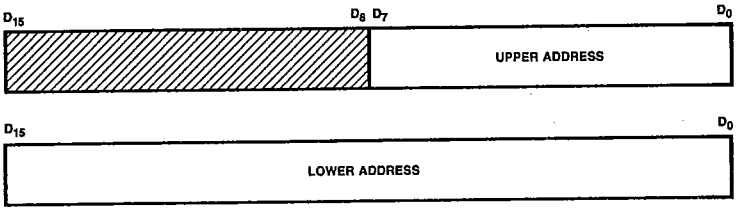
ADDRESS	REGISTER	ACTIVE BITS
0 0 1 0 0	Top of Page Soft (HI)	14 . . . 8
0 0 1 0 1	Top of Page Soft (LO)	15 . . . 0
0 0 1 1 0	Top of Window Soft (HI)	14 . . . 8
0 0 1 1 1	Top of Window Soft (LO)	15 . . . 0
0 1 0 0 1	Top of Page Hard (HI)	14 . . . 8
0 1 0 1 0	Top of Page Hard (LO)	15 . . . 0
0 1 0 1 1	Top of Window Hard (HI)	14 . . . 8
0 1 1 0 0	Top of Window Hard (LO)	15 . . . 0



TOP OF PAGE/TOP OF WINDOW REGISTERS L/$\overline{S}$ = 1

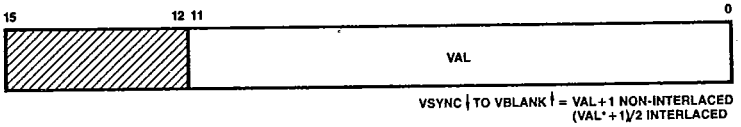
READ/WRITE

ADDRESS	REGISTER	ACTIVE BITS
0 0 1 0 0	Top of Page Soft (HI)	7...0
0 0 1 0 1	Top of Page Soft (LO)	15...0
0 0 1 1 0	Top of Window Soft (HI)	7...0
0 0 1 1 1	Top of Window Soft (LO)	15...0
0 1 0 0 1	Top of Page Hard (HI)	7...0
0 1 0 1 0	Top of Page Hard (LO)	15...0
0 1 0 1 1	Top of Window Hard (HI)	7...0
0 1 1 0 0	Top of Window Hard (LO)	15...0



VERTICAL ACTIVE LINES REGISTER

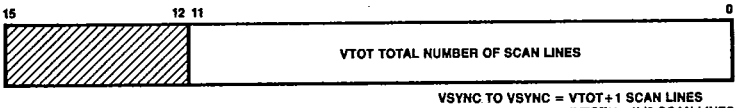
ADDRESS: 1 0 0 1 0
WRITE



VSYNC ↓ TO VBLANK ↓ = VAL+1 NON-INTERLACED
(VAL*+1)/2 INTERLACED

VERTICAL TOTAL LINES REGISTER

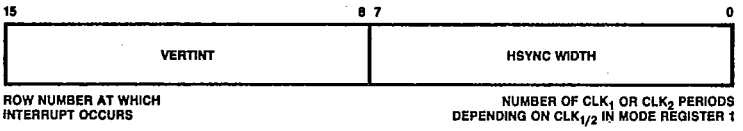
ADDRESS: 1 0 0 1 1
WRITE



VSYNC TO VSYNC = VTOT+1 SCAN LINES
= (VTOT**+1)/2 SCAN LINES

HORIZONTAL SYNC AND VERTICAL INTERRUPT ROW REGISTER

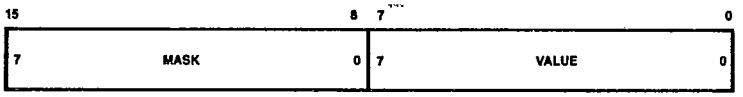
ADDRESS: 1 0 1 0 0
WRITE



*Must be odd
**Must be even

ATTRIBUTE FLAG REGISTER

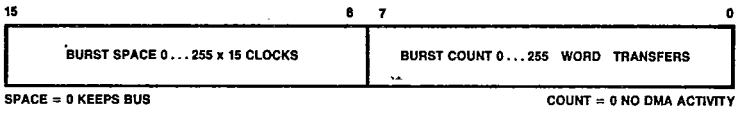
ADDRESS: 0 1 0 0 0
WRITE



Note: When a mask-bit is set to 0, the corresponding value-bit must be 0.

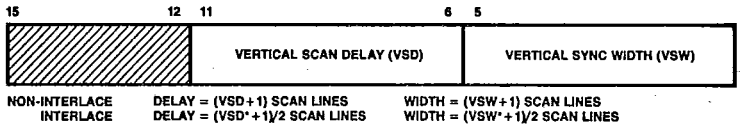
BURST REGISTER

ADDRESS: 1 0 0 0 0
WRITE



VERTICAL SYNC WIDTH/VERTICAL SCAN DELAY REGISTER

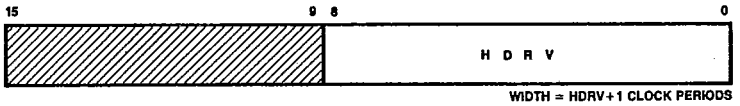
ADDRESS: 1 0 0 0 1
WRITE



*Must be odd

HORIZONTAL DRIVE REGISTER

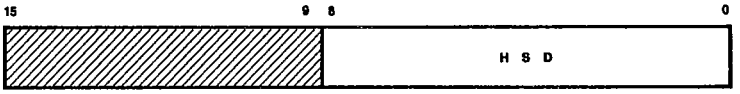
ADDRESS: 1 0 1 0 1
WRITE



WIDTH = HDRV+1 CLOCK PERIODS

HORIZONTAL SCAN DELAY REGISTER

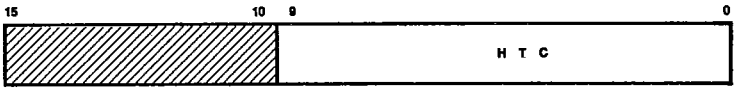
ADDRESS: 1 0 1 1 0
WRITE



DELAY = HSD+1 CLOCK PERIODS

HORIZONTAL TOTAL COUNT REGISTER

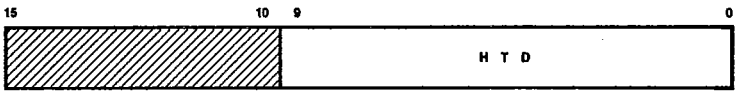
ADDRESS: 1 0 1 1 1
WRITE



HSYNC PERIOD = HTC+1 CLOCK PERIODS

HORIZONTAL TOTAL DISPLAY REGISTER

ADDRESS: 1 1 0 0 0
WRITE



HSYNC TO BLANK INTERVAL = HTD+1 CLOCK PERIODS
(MUST BE ODD IN INTERLACE MODE)

FRAME TIMING SIGNALS SUMMARY:

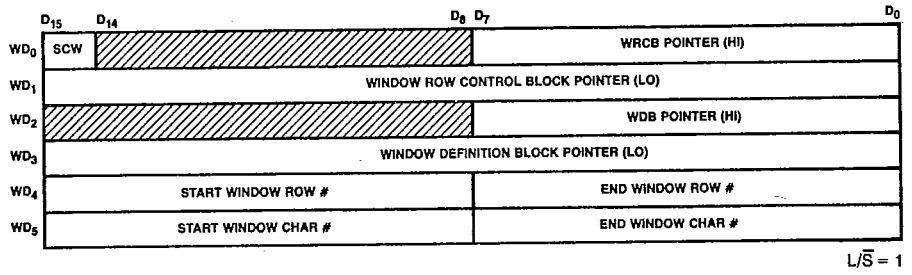
NON-INTERLACED MODE

VERTICAL SYNC WIDTH	VSW + 1
FRONT PORCH (VBLANK R.E. TO VSYNC R.E.)	VTOT − VAL
BACK PORCH (VSYNC F.E. TO VBLANK F.E.)	VSD + 1
VSYNC F.E. TO NEXT VBLANK R.E.	VAL + 1
TOTAL SCAN LINES/FRAME − VSYNC WIDTH	VTOT + 1
HORIZONTAL SYNC WIDTH	HSYNC + 1
HORIZONTAL SYNC PERIOD	HTC + 1
HSYNC R.E. TO NEXT HBLANK R.E.	HTD + 1
HSYNC R.E. TO HBLANK F.E.	HSD + 1
HDRIVE R.E. TO HDRIVE F.E.	HDRV + 1

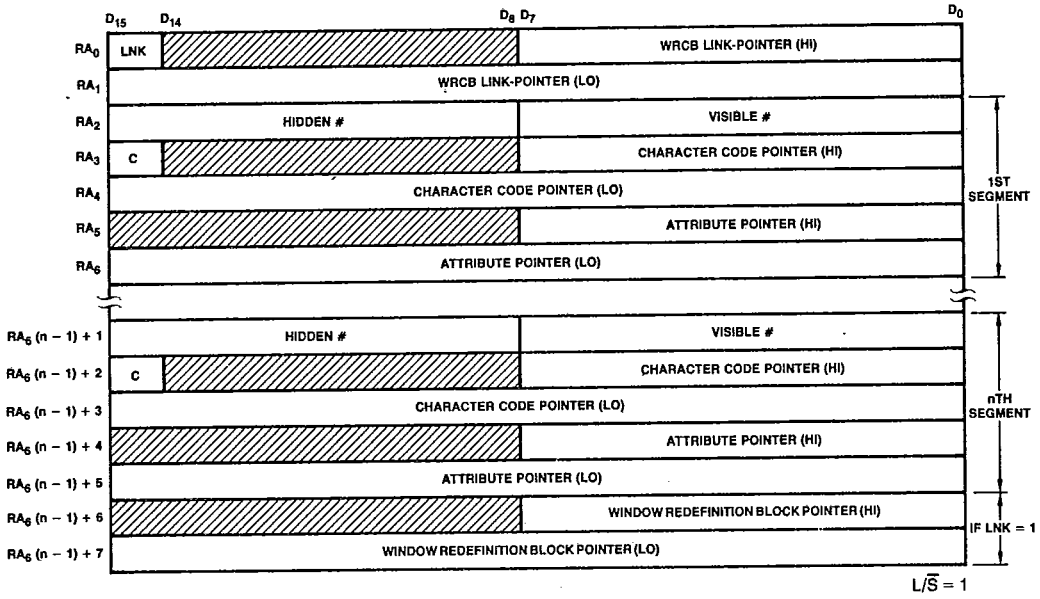
INTERLACED MODE

VERTICAL SYNC WIDTH	(VSW + 1)/2, VSW ODD	
BACK PORCH	VSD/2, EVEN FIELD (VSD + 1)/2, ODD FIELD	VAL ODD
VSYNC F.E. TO NEXT VBLANK R.E.	(VAL + 1)/2, ODD FIELD VAL/2, EVEN FIELD	VSD ODD
TOTAL SCAN LINES/FRAME − VSYNC WIDTH	(VTOT + 1)/2, VTOT EVEN	
HORIZONTAL SYNC WIDTH	HSYNC + 1	
HORIZONTAL SYNC PERIOD	HTC + 1	
HSYNC R.E. TO NEXT HBLANK R.E.	HTD + 1, HTD ODD	
HSYNC E. TO HBLANK F.E.	HSD + 1	
HDRIVE R.E. TO HDRIVE F.E.	HDRV + 1	
FRONT PORCH (VBLANK R.E. TO VSYNC R.E.)	(VTOT − VAL)/2, EVEN FIELD (VTOT + 1-VAL)/2, ODD FIELD VAL ODD, VTOT EVEN	

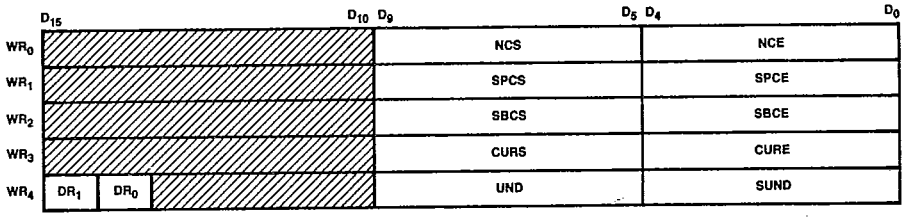
WINDOW DEFINITION BLOCK



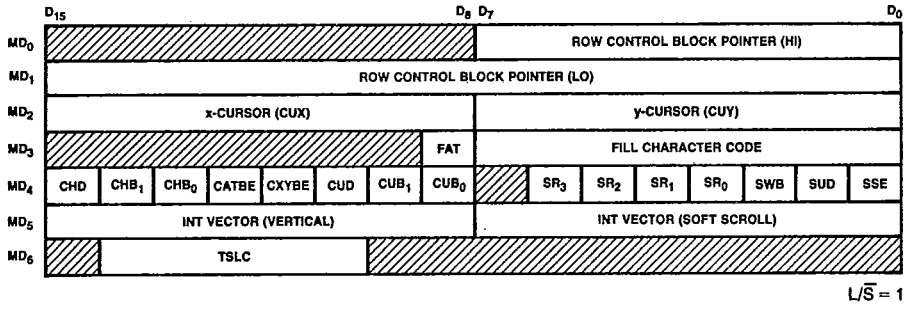
WINDOW ROW CONTROL BLOCK



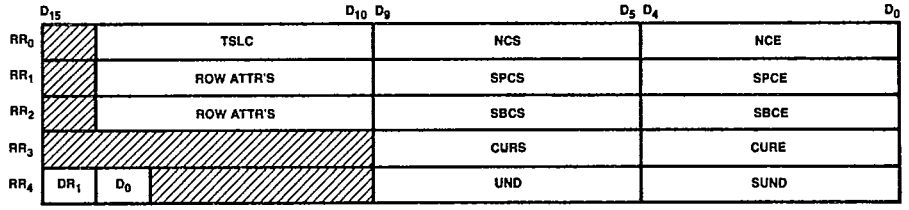
WINDOW REDEFINITION BLOCK



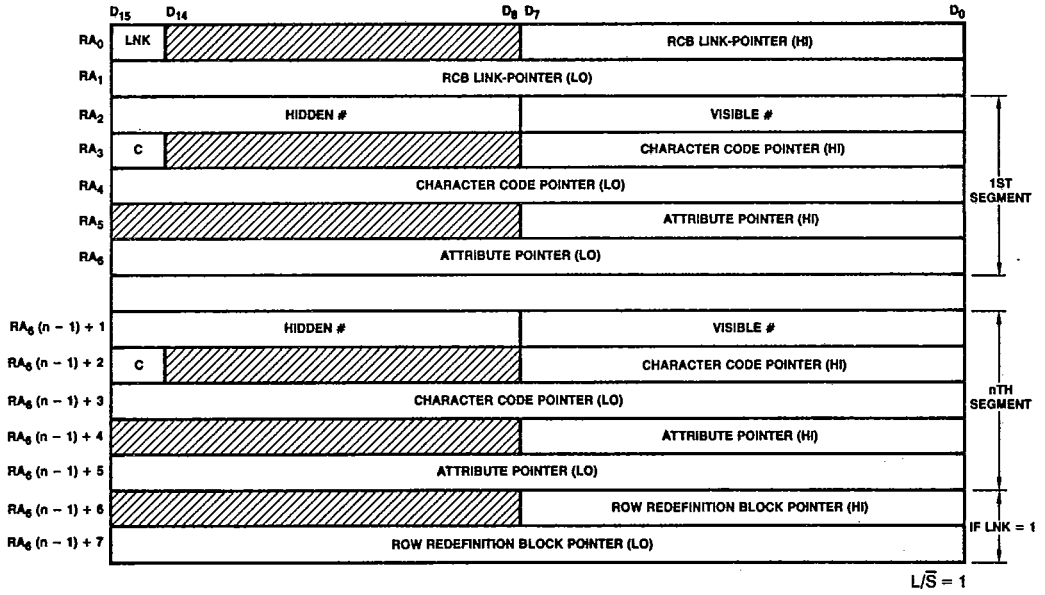
MAIN DEFINITION BLOCK



ROW REDEFINITION BLOCK



ROW CONTROL BLOCK



APPENDIX A

MAXIMUM RATINGS (Above which the useful life may be impaired)

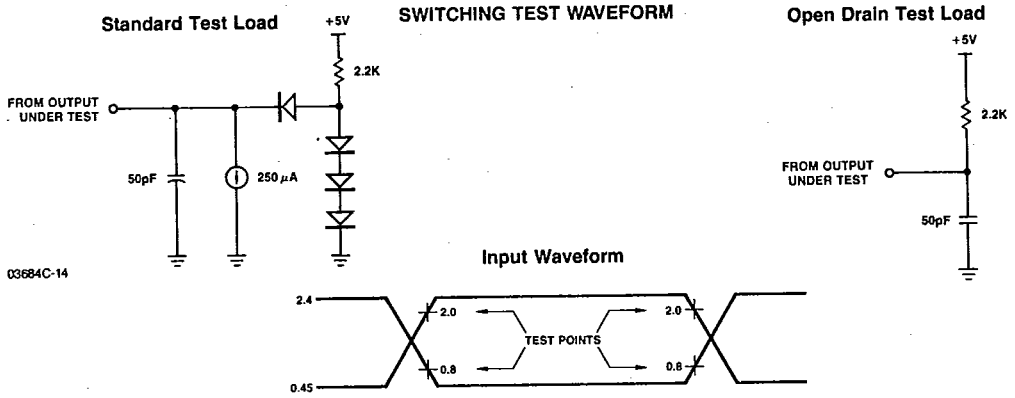
Storage Temperature	−65 to +150°C
Temperature Ambient under Bias	−65 to +125°C
Supply Voltage to Ground Potential Continuous	−0.5 to +7.0V
DC Voltage Applied to Outputs for High Output State	−0.5V to +V_{CC}
DC Input Voltage	−0.5 to +7.0V
DC Output Current into Outputs	30mA
DC Input Current	−30 to +5.0mA

DC CHARACTERISTICS

T_A = 25 to +70°C; V_{CC} = 5V ±5%
C_{LOAD} = 15pF (All inputs except CLK_1 and CLK_2)
C_{LOAD} = 80pF (CLK_1 and CLK_2)

Parameters	Description	Min	Max	Units
V_{OH}	Output High Voltage	2.4		V
V_{OL}	Output Low Voltage (I_{OL} = 3.2mA)	0	0.4	V
V_{IH}	Input High Voltage (except CLK_1 and CLK_2)	2.0	V_{CC} +0.5	V
V_{CIH}	CLK_1/CLK_2 Input High Voltage	V_{CC} −0.5		V
V_{IL}	Input Low Voltage (except CLK_1 and CLK_2)	−0.5	0.8	V
V_{CIL}	CLK_1/CLK_2 Input Low Voltage		0.3	V
I_{IX}	Input Load Current		±10	μA
IO_2, I_O	Output Leakage Current		±10	μA
I_{CC}	Supply Current		500	mA
C_{IN}	Input Capacitance (all pins except CLK_1 and CLK_2)		15	pF
C_{OUT}	Output Capacitance		50	pF
$C_{I/O}$	Bidirectional Capacitance		20	pF

SWITCHING TEST CIRCUITS AND SWITCHING TEST WAVEFORM

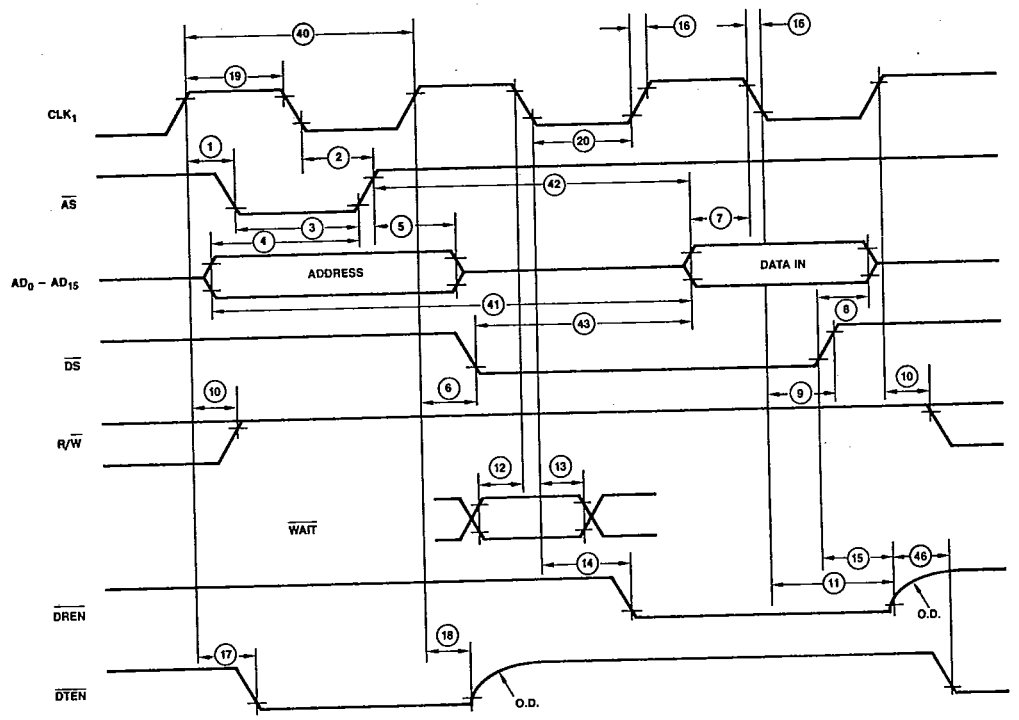


Z8052 BUS MASTER READ/WRITE

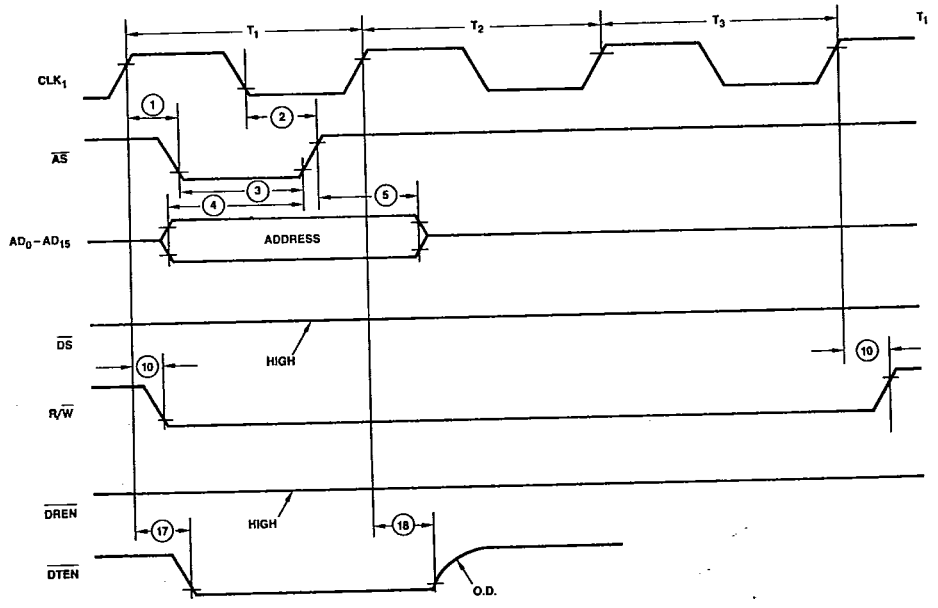
			5MHz		6MHz		8MHz	
			Min	Max	Min	Max	Min	Max
1	t_{PHL}	CLK_1 ↑ TO $\overline{AS}$ ↓		55		55		45
2	t_{PLH}	CLK ↓ TO $\overline{AS}$ ↑		55		55		45
3	t_{PW}	$\overline{AS}$ PULSE WIDTH	60		60		45	
4	t_S	ADDRESS VALID TO $\overline{AS}$ ↑	40		40		30	
5	t_H	ADDRESS FROM $\overline{AS}$ ↑	30		30		30	
6	t_{PHL}	CLK_1 ↑ TO $\overline{DS}$ ↓		65		65		45
7	t_S	DATA IN TO CLK_1 ↓	15		15		10	
8	t_H	DATA IN FROM $\overline{DS}$ ↑	−10		−10		0	
9	t_{PLH}	CLK_1 ↓ TO $\overline{DS}$ ↑		65		65		45
10	t_{PLH}	CLK_1 ↑ TO R/$\overline{W}$	0	55	0	55	0	45
11	t_H	CLK_1 ↓ TO $\overline{DREN}$ ↑		65		65		65
12	t_S	$\overline{WAIT}$ VALID TO CLK_1 ↓	15		15		10	
13	t_H	$\overline{WAIT}$ FROM CLK_1 ↓	20		20		20	
14	t_{PHL}	CLK_1 ↓ TO $\overline{DREN}$ ↓		55		55		45
15	t_H	$\overline{DREN}$ FROM $\overline{DS}$ ↑	−10		−10		0	
16	t_R, t_F	RISE, FALL TIME, CLK_1		15		15		10
17	t_{PHL}	CLK_1 ↑ TO DTEN ↓		55		55		45
18	t_{PLH}	CLK_1 ↑ TO DTEN ↑		55		55		45
19	t_{PW}	CLK_1 HIGH PULSE WIDTH	85	500	70	500	45	500
20	t_{PW}	CLK_1 LOW PULSE WIDTH	75	500	70	500	45	500
40	t_{CYC}	CLK_1 PERIOD	200	1000	165	1000	125	1000
41	t_{AVDV}	ADD VALID TO DATA IN (Note)		300		300		225
42	t_{ASDV}	$\overline{AS}$ ↑ TO DATA VALID (Note)		245		245		185
43	t_{DSDV}	$\overline{DS}$ ↓ TO DATA VALID (Note)		155		155		115
46	t_{DRT}	$\overline{DREN}$ ↑ TO $\overline{DTEN}$ ↓	20		20		20	

(41) = 2 · (40) + (19) − (1) − (3) + (4) − (7) (42) = 2 · (40) − (2) − (7) − (16) (43) = (40) + (19) − (6) − (7) (At max CLK_1 freq.)

Z8052 BUS MASTER READ



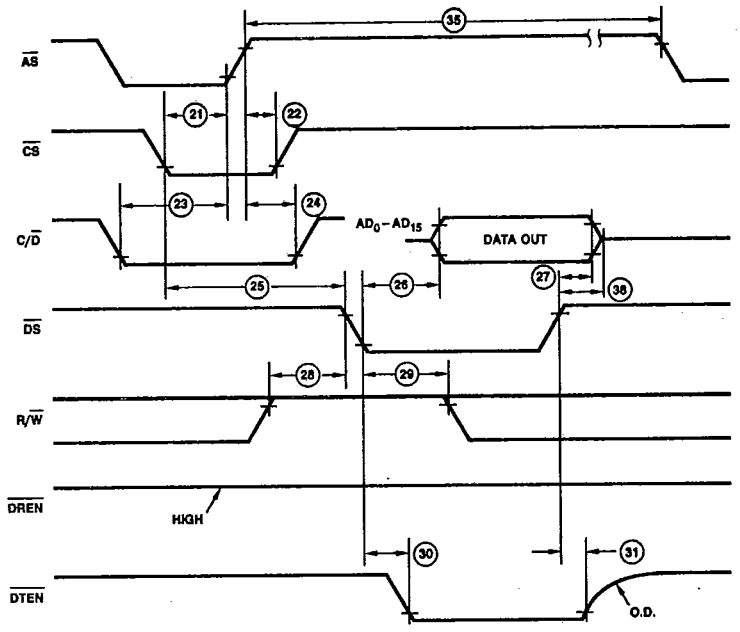
Z8052 BUS MASTER WRITE



Z8052 BUS SLAVE READ LATCHED

			5MHz		6MHz		8MHz	
			Min	Max	Min	Max	Min	Max
21	t_S	$\overline{CS}$ ↓ TO $\overline{AS}$ ↑	0		0		0	
22	t_H	$\overline{CS}$ LOW FROM $\overline{AS}$ ↑	25		25		20	
23	t_S	C/D TO AS ↑	0		0		0	
24	t_H	$\overline{C/D}$ FROM $\overline{AS}$ ↑	25		25		20	
25	t_{PD}	$\overline{CS}$ ↓ TO $\overline{DS}$ ↓	40		40		30	
26	t_{DSDV}	$\overline{DS}$ ↓ TO DATA VALID		150		180		150
27	t_H	DATA VALID FROM DS ↑	10		15		10	
28	t_S	R/$\overline{W}$ TO $\overline{DS}$ ↓	10		10		0	
29	t_H	R/W VALID FROM DS ↑	40		40		40	
30	t_{PD}	DELAY FROM DS		55		55		45
31	t_{PD}	DELAY FROM $\overline{DS}$ ↑ TO $\overline{DTEN}$ ↑		55		55		45
35	t_{SRT}	SLAVE RECOVERY TIME	440		440		330	
38	t_Z	DS ↑ TO AD_0-AD_{15} HI-Z	10	60	10	60	10	50

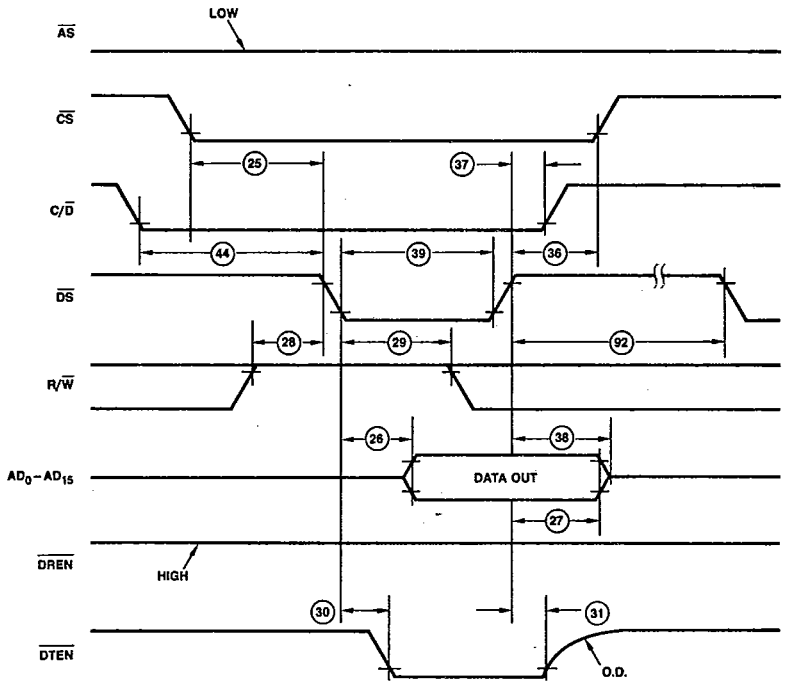
Notes: 1. R/W latched internally by DS ↓.
2. CS latched internally by $\overline{AS}$ ↑.
3. C/D latched internally by $\overline{AS}$ ↑.
4. t_{SRT}: (35) = 3 · (40) − (3)



Z8052 BUS SLAVE READ UNLATCHED

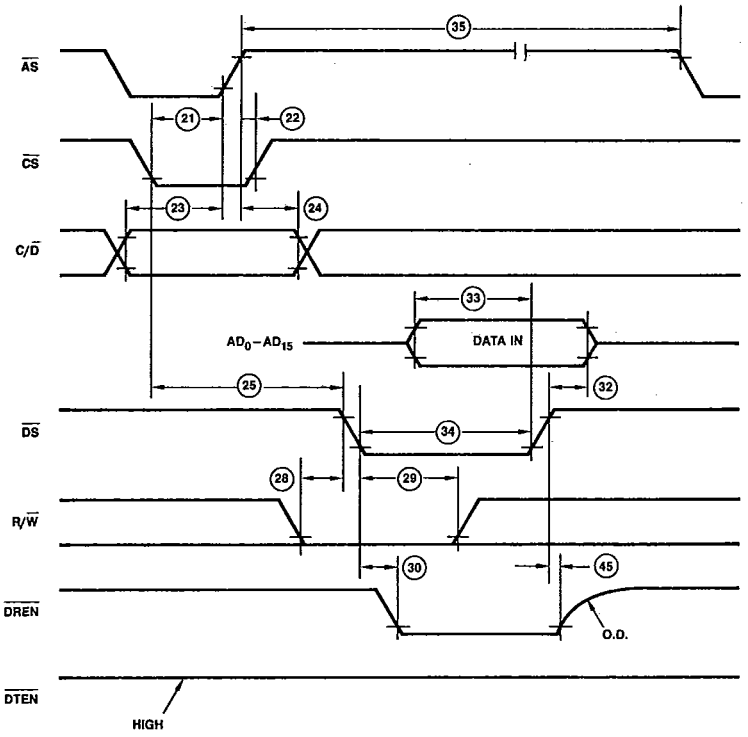
			5MHz		6MHz		8MHz	
			Min	Max	Min	Max	Min	Max
36	t_H	$\overline{CS}$ LOW FROM $\overline{DS}$ ↑	7		7		5	
37	t_H	C/$\overline{D}$ LOW FROM $\overline{DS}$ ↑	7				5	
39	t_{PW}	$\overline{DS}$ ↓ TO DS ↑ READ	200		200		150	
44	t_S	C/D TO $\overline{DS}$ ↓	40		40		30	
92	t_{SRT}	SLAVE RECOVERY TIME (Note 1)	300		300		225	

Note 1: t_{SRT}: (92) = 3 • (40) − (39)



Z8052 BUS SLAVE WRITE LATCHED

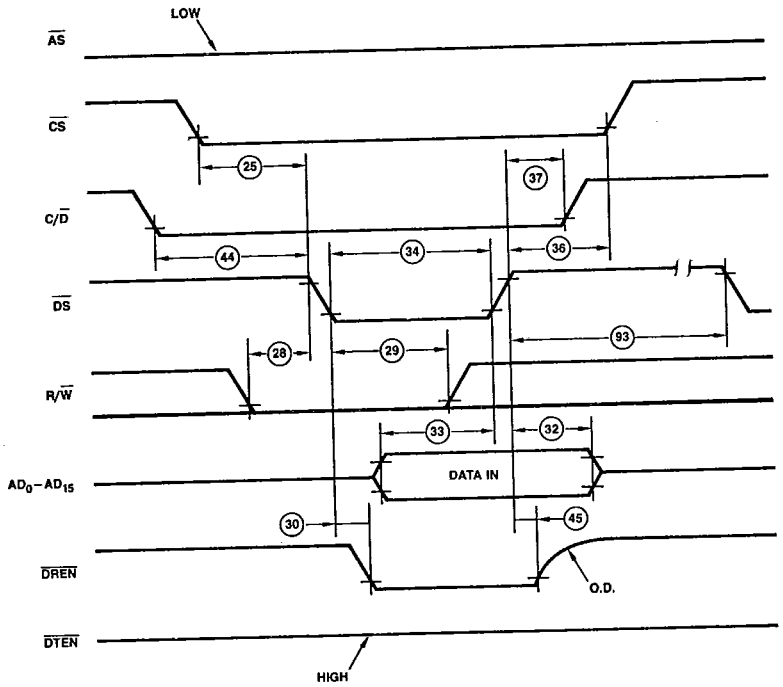
			5MHz		6MHz		8MHz	
			Min	Max	Min	Max	Min	Max
32	t_H	DATA IN VALID FROM $\overline{DS}$ ↑	20		20		20	
33	t_S	DATA IN VALID TO $\overline{DS}$ ↑	90		90		80	
34	t_{PW}	DS PULSE WIDTH	135		135		100	
45	t_H	DELAY FROM $\overline{DS}$ ↑ TO $\overline{DREN}$ ↑	20	70	20	70	20	70



Z8052 BUS SLAVE WRITE UNLATCHED

			5MHz		6MHz		8MHz	
			Min	Max	Min	Max	Min	Max
93	t_{SCT}	SLAVE RECOVERY TIME (Note 1)	365		365		275	

Note 1: t_{SCT}: (93) = 3 • (40) − (34)

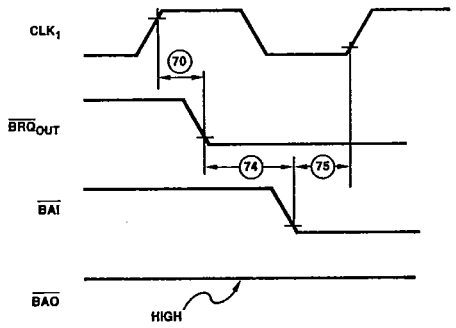


Z8052 BUS EXCHANGE

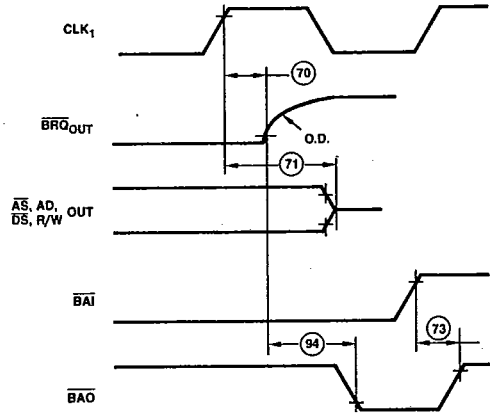
			5MHz		6MHz		8MHz	
			Min	Max	Min	Max	Min	Max
70	t_{PD}	CLK_1 ↑ TO $\overline{BRQ}$ OUT		115		115		100
71	t_{PZ}	CLK_1 ↑ TO FLOAT		115		115		100
72								
73	t_{PD}	$\overline{BAI}$ TO $\overline{BAO}$		50		50		40
74	t_{PD}	$\overline{BRQ}$ ↓ TO $\overline{BAI}$ ↓ DELAY	0		0		0	
75	t_S	$\overline{BAI}$ ↓ TO CLK_1 ↑ (Note 1)	50		50		40	
94	t_{PD}	$\overline{BRQ}$ ↑ TO $\overline{BAO}$ ↓	0	60	0	60	0	50

PRELIMINARY

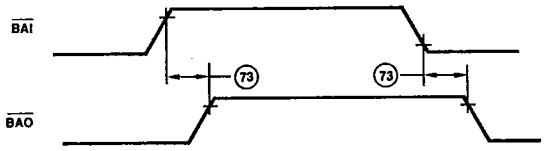
Note 1: This parameter for testing only.



REQUESTING



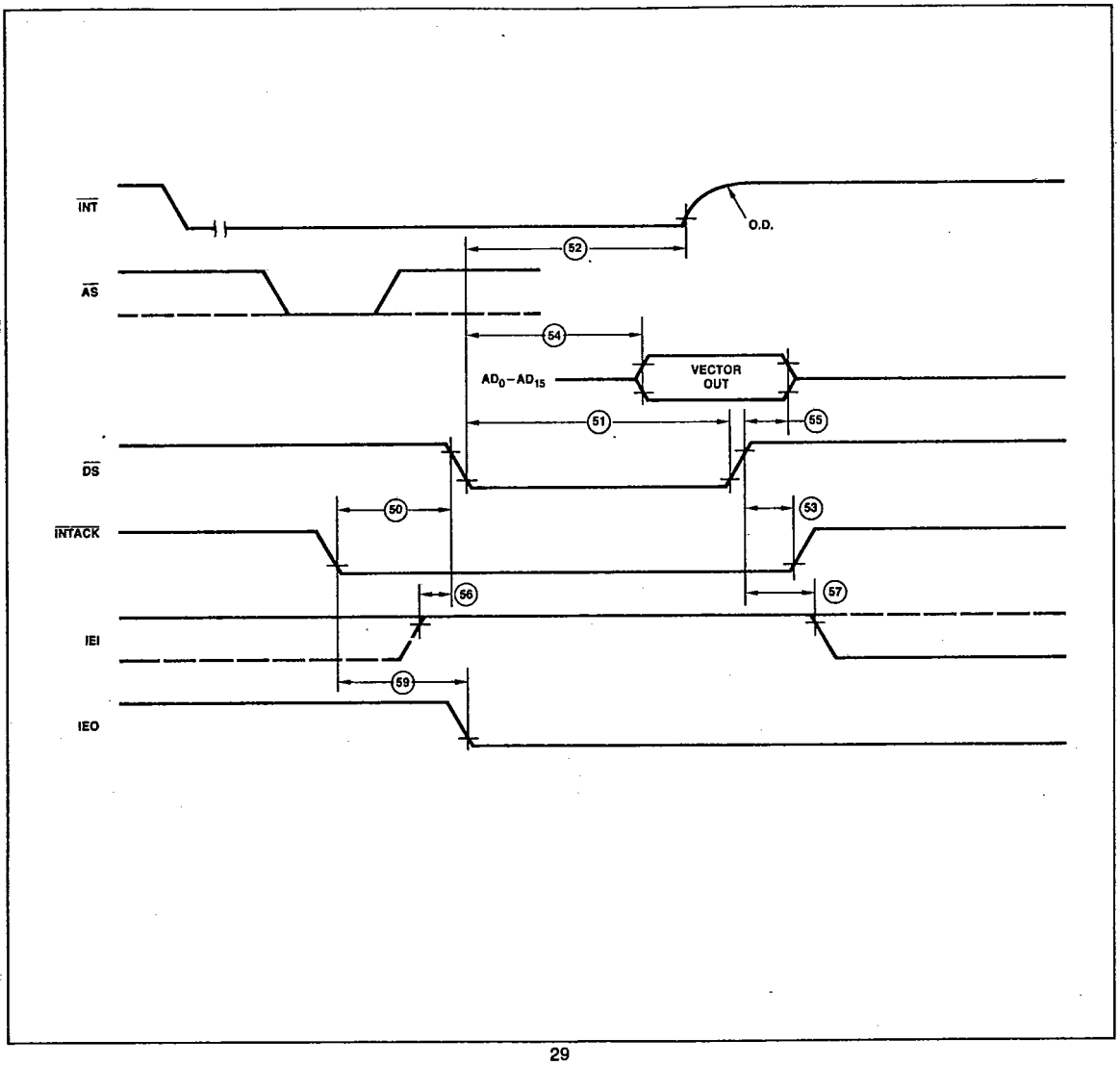
RELEASING



CHAIN DELAY

Z8052 INTERRUPT ACK TIMING — DEVICE ACKNOWLEDGED

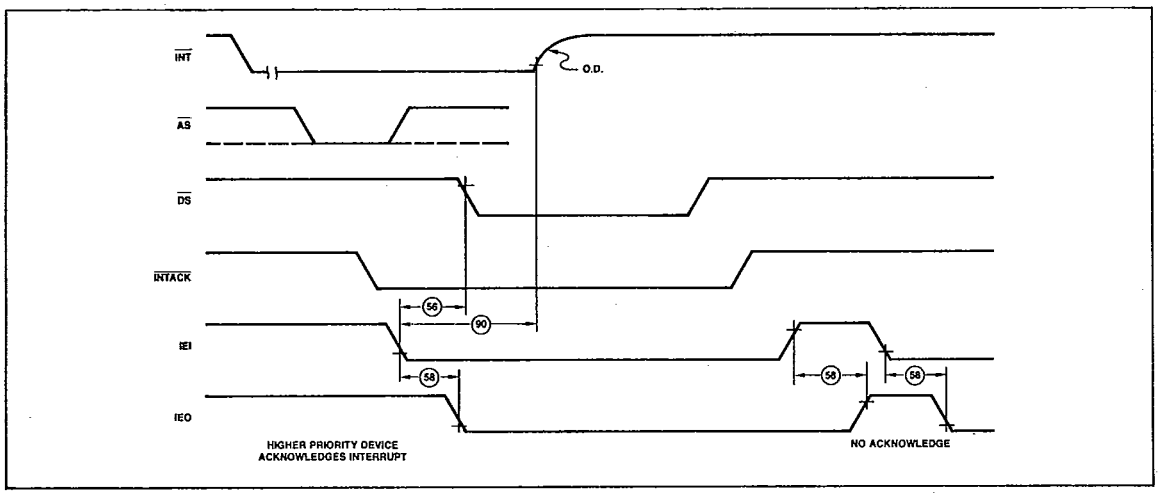
			5MHz		6MHz		8MHz	
			Min	Max	Min	Max	Min	Max
50	t_S	$\overline{INTACK}$ TO $\overline{DS}$ ↓	230		230		200	
51	t_{PW}	$\overline{DS}$ ↓ TO $\overline{DS}$ ↑ ACK	200		200		150	
52	t_{PD}	$\overline{DS}$ ↓ TO INT ↑		230		230		200
53	t_H	$\overline{INTACK}$ FROM $\overline{DS}$ ↑	0		0		0	
54	t_D	$\overline{DS}$ ↓ TO VECTOR VALID		180		180		150
55	t_H	VECTOR FROM $\overline{DS}$ ↑	0		0		0	
56	t_S	IEI TO $\overline{DS}$ ↓	90		90		80	
57	t_H	IEI FROM $\overline{DS}$ ↑	0		0		0	
59	t_D	$\overline{INTACK}$ TO IEO ↓ (IEI = H)		170		170		150



Z8052 INTERRUPT ACK TIMING — LOW PRIORITY

			5MHz		6MHz		8MHz	
			Min	Max	Min	Max	Min	Max
56	t_S	IEI TO $\overline{DS}$ ↓	90		90		80	
58	t_D	IEI TO IEO		90		90		80
90	t_D	IEI ↓ TO $\overline{INT}$ ↑ (Note 1)		90		90		80

Note 1: INT terminated by an acknowledge higher on chain.



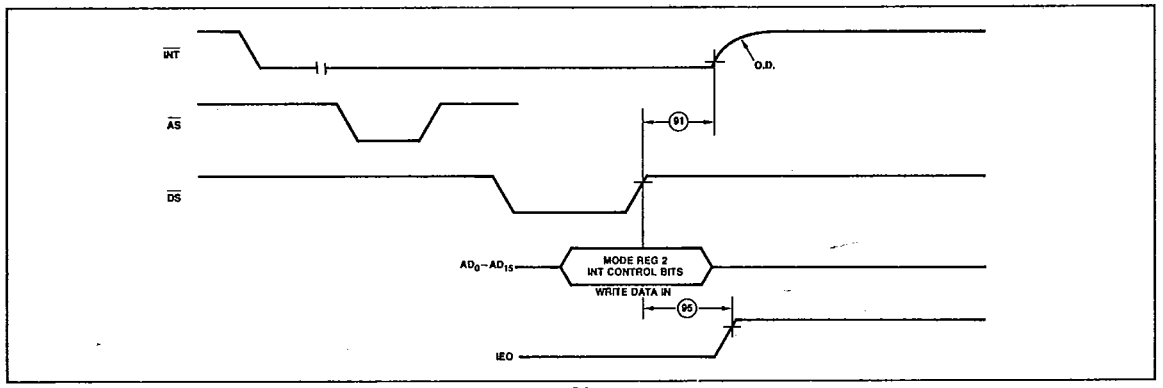
Z8052 NON-VECTORED INT TIMING

			6MHz		8MHz	
			Min	Max	Min	Max
91	t_D	$\overline{DS}$ ↑ TO $\overline{INT}$ (Write) (Note 1)		90		80
95	t_D	$\overline{DS}$ ↑ TO IEO ↑ (Write) (Note 2)		90		80

Notes: 1. This parameter describes the termination of an interrupt request via a write to the appropriate bit in Mode Reg 2;

IUSS	→ 1	IUSV	→ 1
IES	→ 0	IES	→ 0
IPS	→ 0	IPV	→ 0

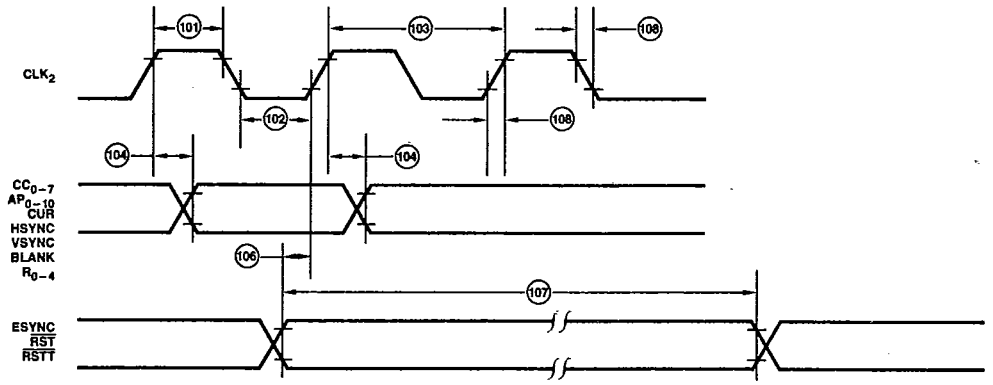
2. This is the release of IEO LOW due to the slave mode reset of the IUS bit in Mode Reg 2.



Z8052 VIDEO OUTPUTS AND SYNCHCRONIZING INPUT TIMING

			5MHz		6MHz		8MHz	
			Min	Max	Min	Max	Min	Max
101	t_{PW}	CLK_2 HIGH PULSE WIDTH	85	500	70	500	35	500
102	t_{PW}	CLK_2 LOW PULSE WIDTH	75	500	70	500	35	500
103	t_{CYC}	CLK_2 PERIOD	200	1000	165	1000	100	1000
104	t_{DC}	CLK_2 ↑ TO OUTPUT DELAY		55		55		35
106	t_S	INPUT SETUP TO CLK_2 ↑ (Note 1)	60		60		50	
107	t_W	INPUT PULSE WIDTH (Note 2)	5T		5T		5T	
108	t_R, t_F	CLK RISE, FALL TIME		15		15		10

Notes: 1. Parameter 106 is specified for test purposes only.
2. Parameter 107 is for reset only. T = CLK_2 period



ORDERING INFORMATION

Z8052 CRTC
68-pin PCC
Z8052-6 VS

Codes

First letter is for package; second letter is for temperature.

C = Ceramic DIP
P = Plastic DIP
L = Ceramic LCC
V = Plastic PCC

R = Protopack
T = Low Profile Protopack
DIP = Dual-In-Line Package
LCC = Leadless Chip Carrier
PCC = Plastic Chip Carrier (Leaded)

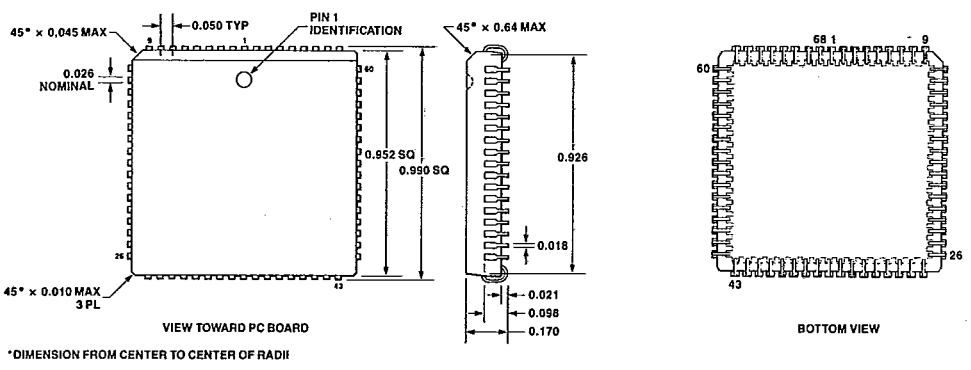
TEMPERATURE
S = 0°C to +70°C
E = −40°C to +85°C
M*= −55°C to +125°C

FLOW
B = 883 Class B

Example: PS is a plastic DIP, 0°C to +70°C.

*For Military Orders, contact your local Zilog Sales Office for Military Electrical Specifications.

PACKAGE INFORMATION



68-Pin Plastic Chip Carrier (PCC)

NOTE: Package dimensions are given in inches. To convert to millimeters, multiply by 25.4.