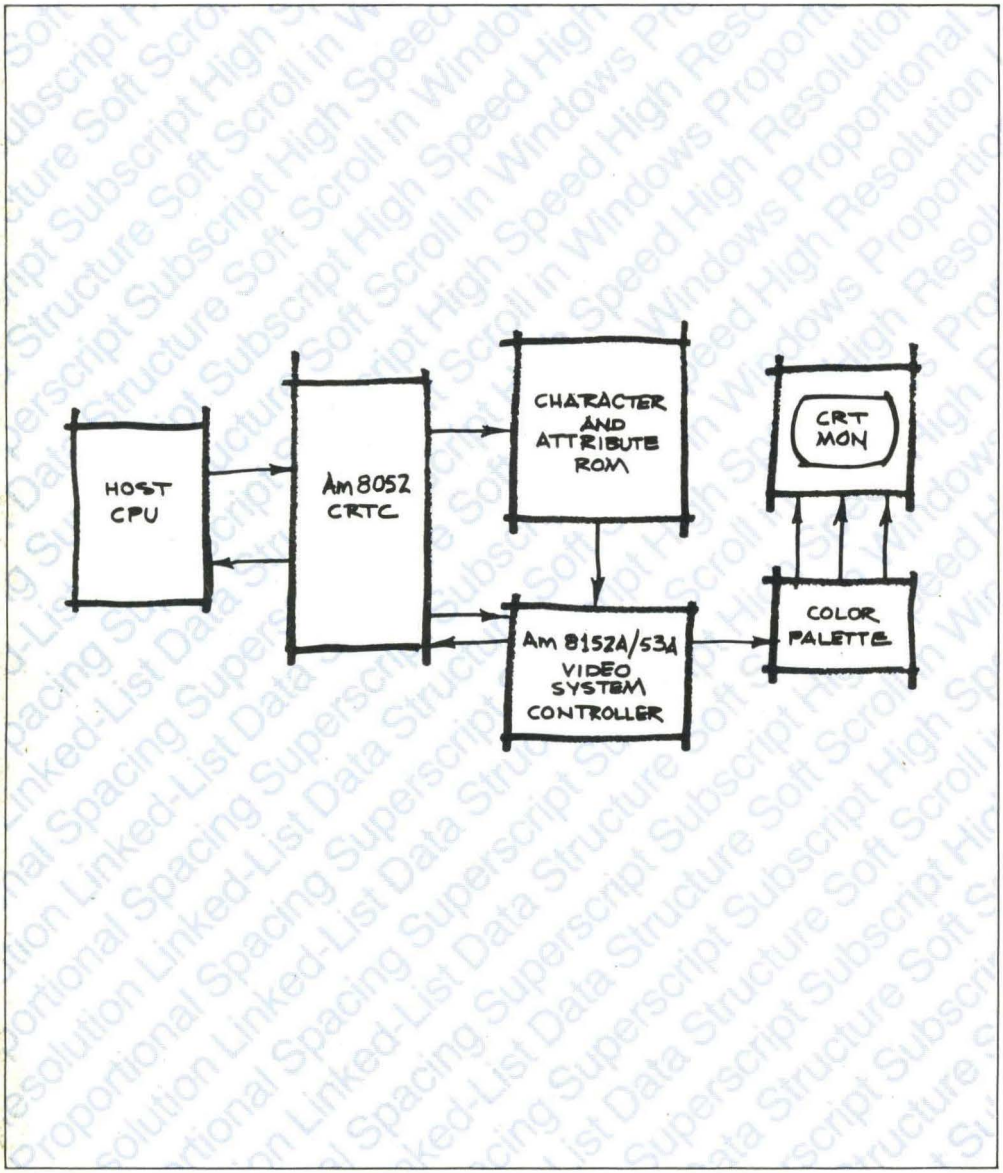
The Am8052 CRT Controller



ADVANCED MICRO DEVICES

Advanced Micro Devices

Am8052 Alphanumeric CRT Controller

Technical Manual

901 Thompson Place, P.O. Box 3453, Sunnyvale, California 94088
(408) 732-2400 TWX: 910-339-9280 TELEX: 34-6306

Printed in U.S.A. CD-B-2.5M-11/84-1

TABLE OF CONTENTS

NOTE: Chapters 1, 3, 4, 5, 6 to be added at a later date.

Acknowledgements:

The bus request/acknowledge and the interrupt daisy-chain mechanisms used in this CRTC are based on the Z8000 structures originated by Zilog, Inc. Figure 2.44 is reprinted with their permission. Despite this conceptual heritage, the Am8052 interfaces very easily to other 16-bit microprocessors.

901 Thompson Place, P.O.Box 3453, Sunnyvale, CA 94088
(408) 732-2400, TWX: 910-339-9280, TELEX: 34-6306

2.1 OVERVIEW

2.1.1 Introduction

The Am8052 CRTC is a general-purpose controller for raster scan CRT displays. It embodies all of the fundamental functions presently provided by various competing monolithic products. In addition, it contains many advanced, next-generation features not presently offered by the competition. A wide variety of systems will be able to take advantage of its features, turning them into powerful display controllers with a minimum chip count.

The Am8052 link-oriented data manipulation provides sophisticated text display without imposing undue overhead on the host CPU.

The versatility of the chip allows it to cover a wide range of applications from medium performance up to very-high performance displays.

2.1.2 General Description

A typical system diagram incorporating the Am8052 CRTC is shown in Figure 2.1.

The circuit can be used together with the Am8152 or Am8153 video display controllers, specially designed to complement the Am8052 and enhance its displaying capabilities.

The Am8052, after initialization by the host processor, acts as a stand-alone device:

It fetches the data to be displayed from the main memory through its internal DMA controller.

It generates the displayable character codes along with their attributes.

It provides all the timing signals to synchronize the beam-scanning with the character-pixel stream.

It configures the screen with useful features such as size programmable windows and vertical soft-scroll.

Due to the sophistication of the Am8052, we will first explain the general operation of the part, splitting it into functional blocks; each block will be described in more detail later.

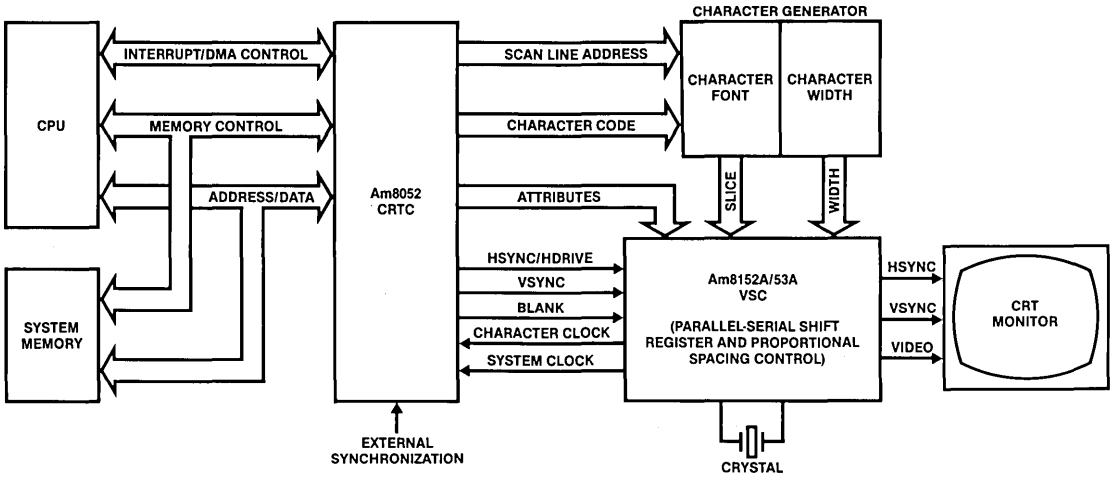


Figure 2.1. CRT Controller System Diagram

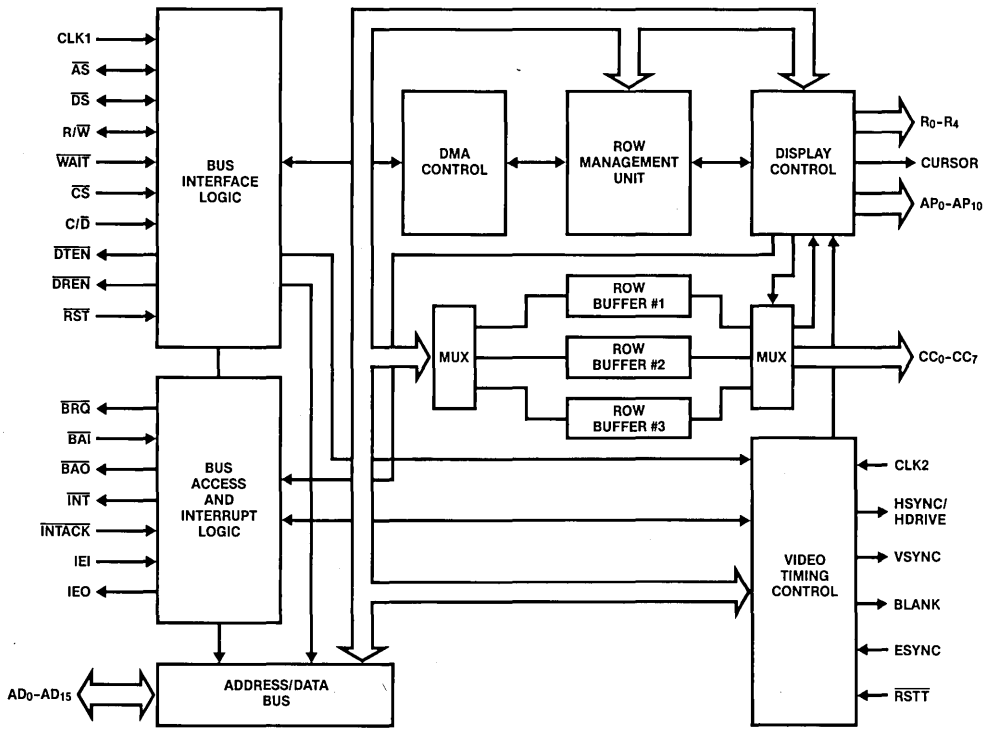


Figure 2.2. Am8052 Block Diagram

The Am8052 is a real time raster scan display controller, which means that it keeps track of updating the display screen on a character-row basis by toggling its internal row-buffers, one being displayed by the display control block, while another is loaded through the DMA interface under control of the row management unit.

All the above operations are synchronized by the video timing control block and intialized by the host processor through the bus interface logic.

The block diagram of the Am8052 (Figure 2.2) shows the functional blocks, and how they interface with each other.

2.1.3 Basic Operation

Following reset, the Am8052 remains in slave mode and waits for the host processor to initialize its timing and control registers, as well as one address pointer to the start of the display data location in the host memory.

While in the idle state, the chip holds both HSYNC and VSYNC signals inactive (LOW), to prevent undefined synchronization to the CRT which might damage high bandwidth tubes, and it also holds the blank signal active to inhibit the CRT beam.

Once the part has been initialized, and upon a command from the CPU, the DMA enters into the bus request sequence to update its three internal row buffers whenever possible. A row buffer cannot be loaded at the same time that it is displayed.

The row management unit governs the loading of the characters to be displayed, as well as their attributes (whenever they are invoked) into the row buffers. This logic is also in charge of updating the display control registers (not accessible to the user), on a row by row basis, as specified by the row definition blocks located in main memory.

Before going further in the description of the internal functionality of the chip, let us see how it handles the data to be displayed and the screen organization.

All the display information (characters, attributes, row structure) reside in main memory in a linked-block structure. Figure 2.3 shows how this structure is implemented with the Am8052.

Before the beginning of each frame display, the CRTC fetches the main definition block from main memory. This block tells it how to set-up the screen-dependent attributes such as soft-scroll rate, character and cursor blink rates and duty-cycles, screen relative cursor coordinates, fill character code, etc.

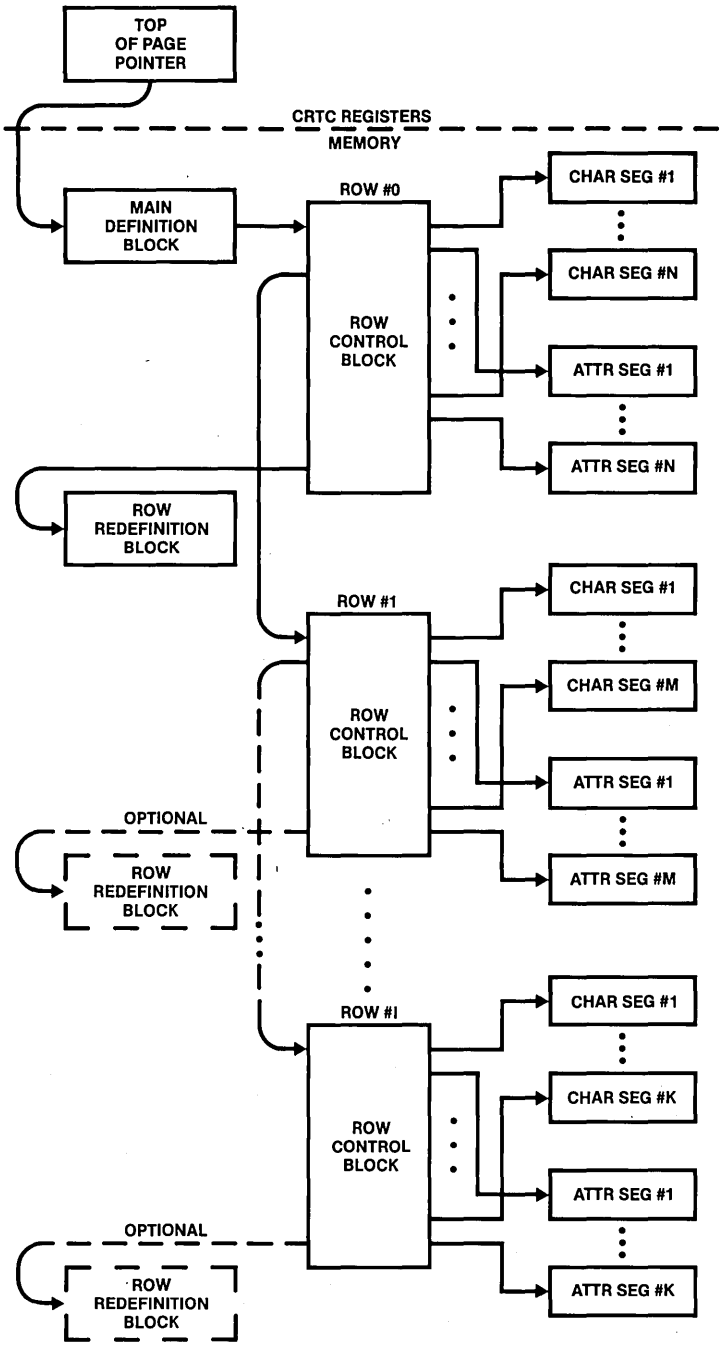


Figure 2.3. Linked-List Architecture 03901A-03

An Am8052 internal register points to the Main Definition Block and links to the first Row Control Block in main memory.

Each Row Control Block links to the next Row Control Block, and contains the pointers to the character and attribute segments to be displayed, both located in main memory. This block also contains the visible and hidden character quantities for each character segment.

Each Row Control Block may optionally invoke a Row Redefinition Block. This Row Redefinition Block allows the user to redefine the row's attributes for the current and succeeding character rows.

The user-defined Row Attributes are contained in a 10-bit code which appears on the attribute bus during the horizontal retrace of the CRT; this is an extra feature which allows the user to define his own row attributes (e.g., invisible row, blinking row, highlight row).

The principal advantages of this linked-list architecture is to decrease the CPU overhead when editing text in system memory. Pointers are manipulated instead of executing CPU intensive data-block moves. Pages are switched on the display by changing a unique pointer (top of page pointer) inside the Am8052. Lines are deleted or inserted by modifying a few pointers in the linked-list structure.

The "display control unit" combines the character stream from one of the three row buffers with the row or character dependent display characteristics of these characters. As a result, the display control block outputs the number of the line being scanned in the current row on R0-R4, and outputs the character-codes contained in this row on CC0-CC7. These two codes form the address sent to the character generator. The character code (most significant bits in the address) points to the matrix of pixels synthesizing the character on the screen, while the line-number (LSB) indicates which line of the matrix is to be displayed on the screen. The resultant line of pixels is output from the character generator and later processed by the various attributes and sent on the video output.

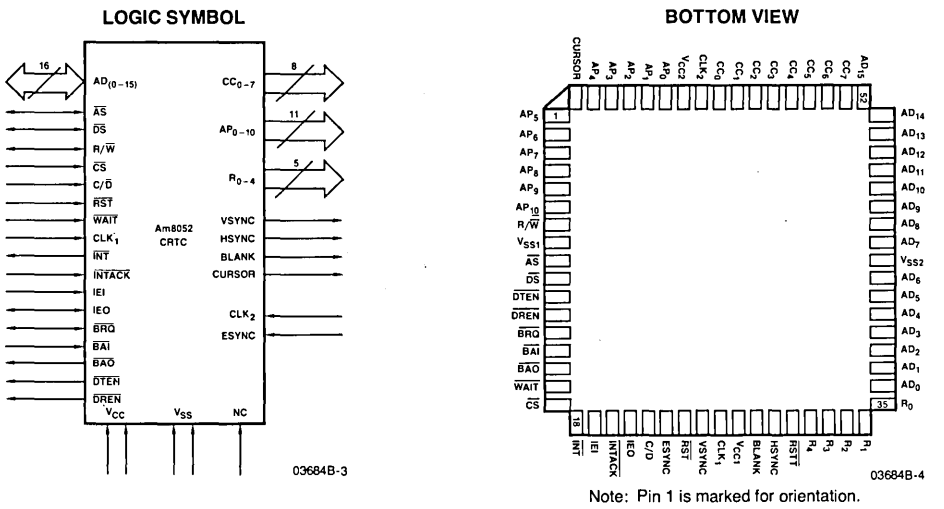


Figure 2.4. Am8052 Pinout

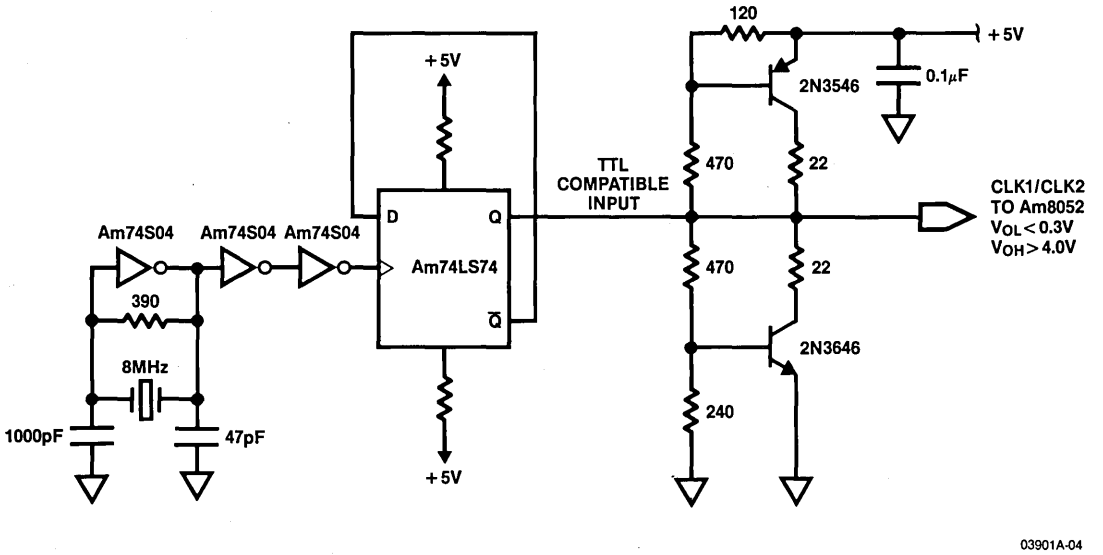


Figure 2.5. CLK1/CLK2 Driver

2.2 INTERFACE SIGNALS

With the exception of CLK1 and CLK2 inputs, all inputs and outputs of the CRTC are TTL compatible. Figure 2.4 shows the device pin-out.

VSS1, VSS2 (Ground)

VCC1, VCC2 (+5V Power Supply)
+/- 5% tolerance for commercial temperature devices.
+/- 10% tolerance for military temperature devices.

CLK1 (System Clock, Input)
The Clock1 signal controls the DMA and peripheral portion of the CRTC, and times all memory accesses. It requires a timing duty cycle of 50% (+/- 5%) at its highest frequency and is driven by an external timing source, usually the CPU clock. In proportional spacing applications, where CLK2 is variable, CLK1 should be used to time the horizontal and vertical sync rates. CLK1 is not TTL compatible. Figure 2.5 shows a CLK1/CLK2 driver generating a clock signal with the required High and Low levels.

CLK2 (Character Clock, Input)
The Clock2 signal is used to time character accesses from the CRTC row buffers. In applications which do not use proportional spacing CLK2 is fixed in frequency and can, therefore, be used to time the horizontal and vertical sync rates, allowing CLK1, the system clock, to be unrelated and asynchronous to the display timing. CLK2 is not TTL compatible.

AD0-AD15 (Address/Data Bus, Input/Output)
The Address/Data Bus is a time-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 bus via its internal DMA capability, it dominates 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 address is output on the ADØ-7 lines. Interrupt Vector information is also output on the ADØ-7 lines.

$\overline{AS}$ (Address Strobe, Input/Output)
Address Strobe is a bidirectional, active Low, three-state signal. In slave mode this input controls the internal transparent latches at the C/$\overline{D}$ and $\overline{CS}$ inputs:

In multiplexed address/data bus systems $\overline{AS}$ High latches C/$\overline{D}$ and $\overline{CS}$.

In demultiplexed address/data bus systems $\overline{AS}$ may be held Low to enable the above-mentioned latches.

When the CRTC is the bus master, $\overline{AS}$ is an output indicating a valid address output on the AD-bus. The address may be latched with the rising edge of $\overline{AS}$. During page address update cycles $\overline{AS}$ and R/$\overline{W}$ are both driven Low.

$\overline{DS}$ (Data Strobe, Input/Output)
Data Strobe is a bidirectional, active Low, three-state signal. When the CRTC is in the slave mode and the external system is transfering information to or from it, $\overline{DS}$ is a timing input used by the CRTC to move data to or from the AD-bus. 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 to or from the AD-bus. For slave mode the $\overline{DS}$ signal may be asynchronous to CLK1.

$\overline{CS}$ (Chip Select, Input)
The $\overline{CS}$ input is an active Low signal used by the host processor to access the CRTC's internal registers. The $\overline{CS}$ input is internally latched through an internal transparent latch controlled by the $\overline{AS}$ input.

$\overline{WAIT}$ (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 only on the falling edge of time T2 of CLK1. $\overline{WAIT}$ is ignored during any non-memory read CRTC cycles.

R/$\overline{W}$ (Read/Write, Input/Output)
Read/Write is a bidirectional, three-state signal. Read polarity is High and write polarity is Low. R/$\overline{W}$ indicates 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. 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 to indicate that the CRTC outputs a high-order address to an external latch. During a dummy DMA cycle, R/$\overline{W}$ is driven High.

C/$\overline{D}$ (Command/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 (C/$\overline{D}$=High) or data (C/$\overline{D}$=Low). When the CRTC is not involved in an I/O transaction with the host, C/$\overline{D}$ is disregarded. C/$\overline{D}$ flows through a transparent latch controlled by $\overline{AS}$. A rising edge of $\overline{AS}$ latches C/$\overline{D}$.

$\overline{\text{DTEN}}$, $\overline{\text{DREN}}$ (Data Transmit Enable, Data Receive Enable, Open Drain Output)
Data Transmit Enable and Data Receive Enable are used to control bus transceivers external to the CRTC, should they be required. When $\overline{\text{DTEN}}$ is Low the transceivers should be driven out from the CRTC onto the bus. When $\overline{\text{DREN}}$ is Low, the transceivers should be driven from the bus into the CRTC. $\overline{\text{DTEN}}$ and $\overline{\text{DREN}}$ are never Low simultaneously.

$\overline{\text{BRQ}}$ (Bus Request, Input/Output, Open Drain)
When the CRTC requires use of the bus for DMA activity, the $\overline{\text{BRQ}}$ line is driven Low. It remains Low until it has ceased using the bus. This pin is also an input pin which allows the CRTC to sense the $\overline{\text{BRQ}}$ line.

$\overline{\text{BAI}}$ (Bus Acknowledge In, Input)
Bus Acknowledge In is an active Low input. When the CRTC requires host bus access and has successfully pulled its $\overline{\text{BRQ}}$ pin Low, a $\overline{\text{BAI}}$ Low input signifies that the CRTC has obtained bus mastership. $\overline{\text{BAI}}$ is internally synchronized for two periods of CLK1 to alleviate metastable problems. When the CRTC does not require host bus access, the $\overline{\text{BAI}}$ input ripples to the $\overline{\text{BAO}}$ output.

When $\overline{\text{BAI}}$ is removed during a DMA burst, the CRTC finishes the current DMA cycle and then releases $\overline{\text{BRQ}}$. If the DMA burst is not completed and no other device requests the bus ($\overline{\text{BRQ}}$ is High), the CRTC asserts $\overline{\text{BRQ}}$ again. The CRTC releases the bus for a minimum of three system clock (CLK1) cycles.

$\overline{\text{BAO}}$ (Bus Acknowledge Out, Output)
$\overline{\text{BAO}}$ output is forced active high when the CRTC has obtained bus mastership, otherwise the $\overline{\text{BAI}}$ input ripples out of the CRTC via the $\overline{\text{BAO}}$ output.

$\overline{\text{INT}}$ (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 $\overline{\text{INTACK}}$ pin.

$\overline{\text{INTACK}}$ (Interrupt Acknowledge, Input)
When this line is driven Low, the CRTC examines its IEI line to determine if it has been granted an acknowledge by the CPU. $\overline{\text{INTACK}}$ must be High for normal operation.

IEI (Interrupt Enable-In, Input)
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. See the section on Interrupt protocol for a full discussion.

HSYNC (Horizontal Sync, Output)
HSYNC is an active High output which controls the horizontal retrace of the CRT's electron beam. This output is held inactive (Low) when the CRTC is reset to prevent unknown synchronization of the CRT which might 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 which controls the vertical retrace of the CRT's electron beam. This output is held Low when 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. It is a composite of horizontal and vertical blank. This output is held active High when the CRTC is reset.

ESYNC (External Sync, Input)
This pin is the external synchronization input and should be used exclusively for power line synchronization. The ESYNC input cannot synchronize two video systems since HSYNC is not altered by this signal. This input is enabled by setting the ES bit in Mode Register 1. For further details on ESYNC operation, refer to Section 2.9.

$\overline{\text{RSTT}}$ (Test Reset, Input)
This pin is used for test purposes. It may, however also be used to synchronize two Am8052 parts for some specific application. Whenever $\overline{\text{RSTT}}$ input is lowered, the following will take effect:

HSYNC is held Low
VSYNC is held Low
Blank is held High
Mode Register 2 bit 0 through bit 8 are reset
Horizontal counter is reset
Vertical counter is reset

For synchronizing two CRTCs, $\overline{\text{RSTT}}$ should be driven synchronously to the video timing clocks (CLK1 or CLK2).

$\overline{\text{RST}}$ (Reset, Input)
A Low on this input for at least 5 clock cycles is interpreted by the CRTC as a reset signal. The effect of reset is to drive all CRTC bus signals into the high-impedance state and initialize Mode Registers 1 and 2. Additionally, the CRTC will go into Slave mode within the defined reset widths.

CCØ-7 (Character Code Outputs)
CCØ-7 outputs are active High. This character port outputs eight bits of data stored in the character code section of the row buffer currently being displayed. The character code output can be delayed by one or two clock periods (CLK2) in order to allow the attribute bits associated with the particular character code to be masked and decoded and to generate suitable synchronized attribute control.

RØ-4 (Scan Line Control, Outputs)
RØ-4 outputs are active High. These outputs carry the binary address of the character slice being displayed. These outputs form the least significant address portion of a character ROM. The outputs are held High for the scan lines that do not carry active video.

APØ-AP1Ø, (Attribute Port, Outputs)
These eleven lines output the attribute information associated with the characters. During HSYNC the row attribute word contained in the Row Redefinition Block is output on APØ-AP4 and AP6-AP1Ø. This word can be stored externally by the falling edge of HSYNC.

CURSOR (Cursor, Output)
This pin is the cursor output indicator. See character attributes section for further information.

POINTER ADDRESS (AD4-AD0)

HEX	TYPE	ACTIVE BITS	REGISTER NAME
ØØ	R/W	16	MODE 1
Ø1	R/W	16	MODE 2
Ø2	W	12	ATTRIBUTE ENABLE
Ø3	W	5	ATTRIBUTE REDEFINITION
Ø4	R/W	8	TOP OF PAGE SOFT (HI-ORDER)
Ø5	R/W	16	TOP OF PAGE SOFT (LO-ORDER)
Ø6	R/W	8	TOP OF WINDOW SOFT (HI-ORDER)
Ø7	R/W	16	TOP OF WINDOW SOFT (LO-ORDER)
Ø8	W	16	ATTRIBUTE FLAG
Ø9	R/W	8	TOP OF PAGE HARD (HI)
ØA	R/W	16	TOP OF PAGE HARD (LO)
ØB	R/W	8	TOP OF WINDOW HARD (HI)
ØC	R/W	16	TOP OF WINDOW HARD (LO)
1Ø	W	16	DMA BURST
11	W	12	* VSYNC WIDTH/SCAN DELAY
12	W	12	* VERTICAL ACTIVE LINES
13	W	12	* VERTICAL TOTAL LINES
14	W	16	* HSYNC/VERTINT
15	W	9	* HDRIVE
16	W	9	* H SCAN DELAY
17	W	10	* H TOTAL COUNT
18	W	10	* H TOTAL DISPLAY

Table 1: Am8052 Registers

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

2.3 REGISTER DESCRIPTION

2.3.1 Introduction

This chapter provides a brief description of the command, status, and screen configuration registers contained in the CRTC. Each description includes the register address, the operation of the individual register fields and the state of the register after a reset (hardware or software).

Table 1 is a summary of the CRTC's 22 registers. The registers are addressed by an internal pointer which is five bits wide. The pointer is loaded via AD0-AD4 on the external AD-bus with a slave mode write and C/$\overline{D}$ input High.

For details of the addressing schemes see in Section 2.3.2.

To set up the registers: 1) clear the DE bit of Mode Register 1 by hardware reset or by loading the register, 2) initialize all registers except Mode Register 1 with the appropriate values, 3) load Mode Register 1 with the DE-bit set to enable the display, and 4) load Mode Register 2.

Addressing the CRTC with non-specified pointers (ØØ-OF, 19-1F) will cause no problems. The registers can be loaded using a simple software loop starting at 00 and ending at 1F.

2.3.2 Register Addressing

The registers can be accessed only when the CRTC is in the slave mode. They are addressed in a two-step sequence:

1. Pull $\overline{CS}$ Low and C/$\overline{D}$ High to indicate a command type cycle, and strobe the register address present on ADØ-AD4 with an active Low pulse on $\overline{DS}$; this clocks the register address into an internal pointer register. The pointer value remains valid until changed.

2. Reaccess the CRTC with $\overline{CS}$ Low and C/$\overline{D}$ Low to read or write the register pointed by the internal pointer. The data is strobed in or out by the $\overline{DS}$ signal.

The CRTC is in slave mode if it has not been granted control of the bus. Even if the CRTC has asserted $\overline{BRQ}$, it is still in slave mode until it receives an acknowledge and takes over bus control. The CPU can access the CRTC registers any time, i.e. the CRTC places no restrictions on slave accesses.

CRTC SLAVE TRANSFERS:
All slave transfers with the CRTC can be carried out asynchronously with respect to the CRTC CLK1 input. Only $\overline{AS}$ and $\overline{DS}$ are used to time the information transfer. All transfers require two separate bus transactions. First, a pointer value must be written into the CRTC. The pointer value is written by carrying out a slave write transaction

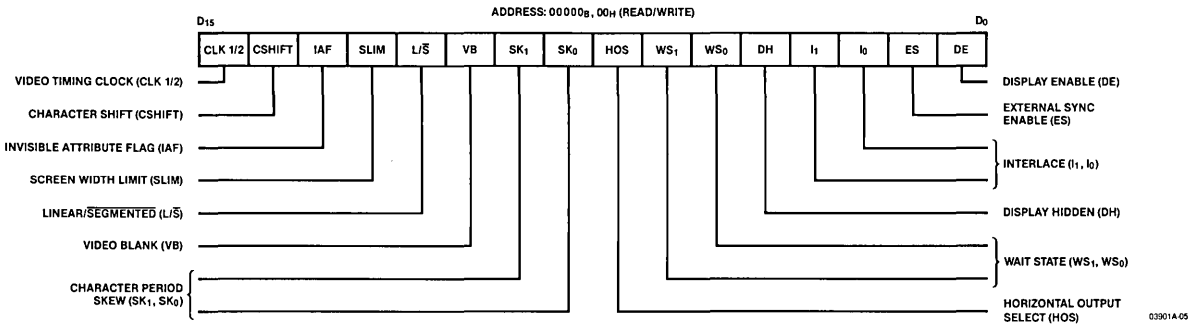


Figure 2.6. Mode Register 1

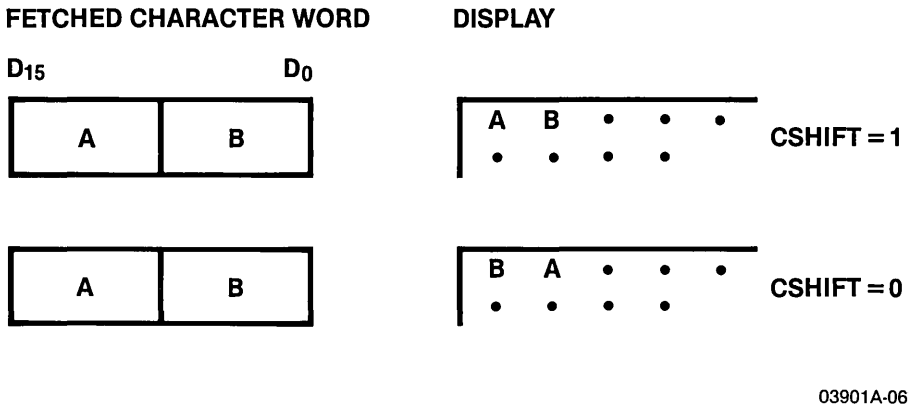


Figure 2.7. Character Positioning on the Display as a Function of CSHIFT

with C/$\overline{D}$ pulled High. Following the setting of the pointer, the data transfer can take place between the bus master and the CRTC internal register designated by the pointer.

The slave transaction typically commences with a pointer write, although repetitive accesses to the same CRTC register can be made without any intervening pointer modification. The transaction is timed off the $\overline{DS}$ signal, since $\overline{AS}$ may not be present in certain systems. The read transaction commences from the leading edge of $\overline{DS}$. The write transaction takes place on the trailing edge of $\overline{DS}$. The $\overline{AS}$ input is used to drive a transparent latch on the CRTC, which is used to capture C/$\overline{D}$ and $\overline{CS}$ in a multiplexed address/data system. If the system is demultiplexed, then $\overline{AS}$ should be driven Low when the CRTC is in the slave mode. This will drive the latch permanently transparent, allowing the demultiplexed $\overline{CS}$ and C/$\overline{D}$ to pass into the CRTC. When the $\overline{DS}$ goes Low and a read transaction is in progress, the CRTC drives the read data onto its ADØ-15 lines and also drives $\overline{DTEN}$ Low. This enables any off-chip bus transceivers, allowing the data to be transmitted to the bus master. When the bus master captures the data, it drives the $\overline{DS}$ signal High. This causes the CRTC to cease driving its ADØ-AD15 lines and also causes $\overline{DTEN}$ to return High, switching off the bus transceivers.

2.3.3 Register Description

MODE REGISTER 1:

The Mode Register 1 contains display and DMA control bits (Figure 2.6). When the device is reset, all Mode Register 1 bits are cleared.

VIDEO TIMING CLOCK--CLK1/2 (D15)
This bit indicates which of CLK1 and CLK2 inputs will be used to time HSYNC (or HDRIVE), VSYNC and VIDEO-BLANK outputs. In proportional spacing applications CLK1 usually times the sync signal, since the frequency of CLK2 is modulated by the character width.

CLK1/2=Ø: CLK2 will be selected for clocking the sync counters
CLK1/2=1: CLK1 will be selected for clocking the sync counters

CHARACTER SHIFT--CSHFT (D14)
This bit allows the CRTC to interpret a two-byte word fetched from main memory, in either of two ways, according to the system's 16-bit word organization. This applies only to characters since they are the only byte-data the CRTC has to interrupt (Figure 2.7). Hence, 16-bit word data such as addresses, attributes, etc., remain unaffected by CSHFT.

INVISIBLE ATTRIBUTE FLAG--IAF (D13)
IAF=Ø: The character that invoked an attribute is loaded into the row buffer, and subsequently displayed. The character is affected by the attribute word.

IAF=1: The character that invoked an attribute is not loaded into the line buffer, and is not displayed.

SCREEN WIDTH LIMIT--SLIM (D12)
The SLIM bit controls the number of characters loaded in each row buffer to either 132 or 96. This can reduce bus overhead when the CRTC row length is 96 characters or less.

SLIM=Ø: The row buffer size is set to 132 characters.
SLIM=1: The row buffer size is set to 96 characters.

LINEAR/SEGMENTED MODE--L/S (D11)
This bit indicates whether the system/display memory access is accomplished by addressing it in a linear or segmented mode.

L/$\overline{S}$=Ø: The CRTC is set for segmented addressing. The linked-list address pointers are two words long. Seven bits (8:14) of the first word define the segment address. The second 16-bit word is the offset address within the segment. Any overflow of the 16-bit offset address does not carry into the upper 7-bit segment address.

L/$\overline{S}$=1: The CRTC is set up for a linear addressing scheme. The most significant byte of the 24-bit linear address is stored in the lower half of the first word (Ø:7). The second word holds the remaining 16 bits.

During page update cycles the CRTC puts out the upper part of the 23/24-bit address on ADØ-AD7. The user may latch the 7/8-bit address.

VIDEO BLANK--VB (D1Ø)
This bit allows the user to blank the screen while making changes in the displayed text or when switching the context. The linked list must, however, be valid before VB is reset.

VB=Ø: Normal Operation
VB=1: The horizontal and vertical sync circuitry and outputs operate normally and the blank output is forced active High. DMA operation is suspended - normal operation resumes when VB=Ø and the next vertical blanking period occurs.

CHARACTER PERIOD SKEW--SK1, SKØ (D9, D8)
The skew bits are useful when implementing custom video systems which involve a significant amount of external character processing preceeding their display. In such systems the processed character may be logically delayed with respect to the VSYNC, HSYNC and BLANK signals. The skew bits program various delays in number of character clock cycles applied to the VSYNC, HSYNC and BLANK signals with respect to character code output. The attributes and cursor outputs can also be selectively delayed by SKØ and SK1. The following combinations are programmable:

BIT SETTINGS		SIGNAL SKEW (# OF CLK2 CYCLES)		
SK1	SKØ	HSYNC,VSYNC AND BLANK	AP0-10 AND CURSOR	CC0-7 AND R0-4
0	0	0	0	0
0	1	1	0	0
1	0	2	1	0
1	1	1	1	0

HORIZONTAL OUTPUT SELECT--HOS (D7)

HOS=Ø: The HSYNC/HDRIVE output pin outputs the horizontal sync timing as programmed in the HSYNC register.

HOS=1: The HSYNC/HDRIVE output pin outputs the horizontal drive timing as programmed in the HDRIVE register.

WAIT STATE--WS1, WSØ (D6, D5)

These bits indicate the number of Wait states inserted for each DMA cycle. These Wait states are in addition to any externally applied wait states. When the CRTC is in Slave Mode, these bits are ignored.

WS1	WSØ	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

DISPLAY HIDDEN--DH (D4)

Applies only to characters associated with Ignore Attribute:

DH=Ø: Those characters are not loaded into the row-buffer.

DH=1: Those characters are treated as displayable information. (See section 2.6.1, Ignore attribute.)

INTERLACE--I1, IØ (D3, D2)

Control the timing of non-interlaced, interlaced, repeat field interface video to support different CRTs (see Section 2.10).

I1	IØ	MODE OF OPERATION
0	0	Non-Interlaced Video
0	1	Reserved
1	0	Repeat Field Interlace (RFI)
1	1	Interlaced Video

EXTERNAL SYNC ENABLE--ES (D1)

Enables the ESYNC input for power line synchronization.

ES=Ø: ESYNC input is ignored

ES=1: A rising edge at the ESYNC input during a vertical-retrace active period (even frame only in interlaced mode) causes the HSYNC/DRIVE output to go (or remain) active for a full horizontal retrace period. The VSYNC active period is stretched, even when register timing signifies an end to vertical retrace, until an ESYNC falling edge occurs.

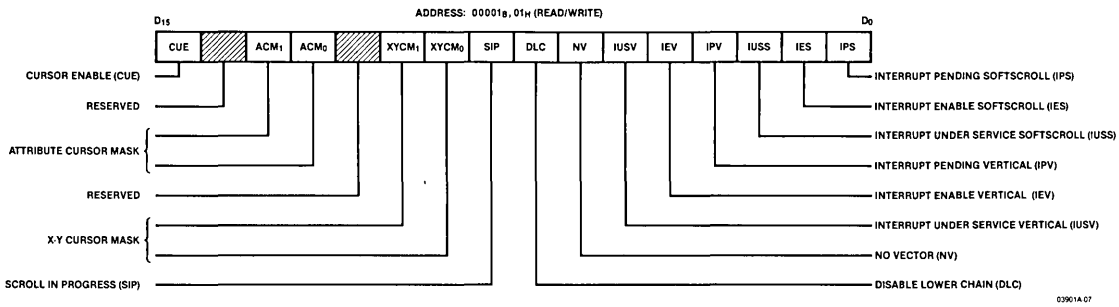


Figure 2.8. Mode Register 2

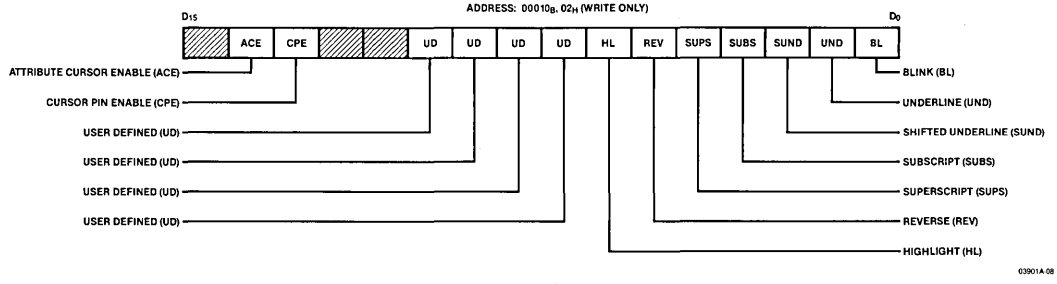


Figure 2.9. Attribute Port Enable Register

DISPLAY ENABLE--DE (D0)

DE=Ø: VSYNC, HSYNC/HDRIVE outputs are inactive and the BLANK output is held active. DMA operation is disabled. The DE bit is reset by the the host processor command or by chip reset ($\overline{RST}$=Low). DE=Øsets all scroll logic to the non-scrolling state.

Resetting DE bit is equivalent to pulling the $\overline{RST}$ pin LOW causing the CRTC to enter the reset procedure.

DE=1: CRTC display operation enabled. DE is set by a host processor command. Setting the DE=1 causes the VSYNC, HSYNC, and BLANK outputs to become active and the DMA controller on board the CRTC eventually requests access to the system bus.

MODE REGISTER 2:

The Mode Register 2 contains the primary control bits for the interrupt control logic, and cursor definition (Figure 2.8).

Upon reset, all Mode Register 2 bits are reset to zero.

CURSOR ENABLE--CUE (D15)

CUE=Ø: The CRTC does not output any XY cursor information.

CUE=1: The cursor X and Y register is enabled. CRTC outputs cursor at the character position defined by the XY cursor register (see Main Definiton Block).

ATTRIBUTE CURSOR MASK - ACM1, ACMØ, (D13, D12)
XY CURSOR MASK - XYCM1, XYCMØ (D1Ø, D9)
The cursor mask field (D13, D12, D10, D9) defines the type of cursor that will be generated when a cursor is required. This field is divided into two parts:

D13	D12	CURSOR ATTRIBUTE DEFINITION
0	0	Cursor Pin Whole
0	1	Cursor Pin Part
1	0	Underline
1	1	Reverse

D10	D9	XY CURSOR DEFINITION
0	0	Cursor Pin Whole
0	1	Cursor Pin Part
1	0	Underline
1	1	Reverse

"Cursor Pin Whole" means that the cursor will be output on the cursor pin for every scan line of that character position (TSCC). CURS and CURE of the Row Definition Block are ignored.

"Cursor Pin Part" means that the cursor will be output on the cursor pin for those scan lines specified in the Row Redefinition Block (CURS and CURE).

"Underline" (BLOB) means that the cursor is output on the underline pin (AP1) for the scan lines specified in the Row Redefinition Block (CURS and CURE).

"Reverse" (part) means that the cursor is output on the reverse pin (AP5) for the scan lines specified in the Row Redefinition Block (CURS and CURE).

SCROLL IN PROGRESS--SIP (D8)
SIP is a status bit that is set/reset by the CRTC smooth scroll control logic.

SIP=Ø: The CRTC is not currently scrolling.
SIP=1: The CRTC is scrolling either window or background.

DISABLE LOWER CHAIN--DLC (D7)

DLC=Ø: IEO operates normally.

DLC=1: The Interrupt Enable Out (IEO) output of the device is forced Low, disabling interrupts from all lower priority devices on the daisy-chain.

NO VECTOR--NV (D6)

NV=Ø: The CRTC outputs the interrupt vector programmed in the Main Definition Block (See the section on Main Definition Block and Interrupt.)

NV=1: The device is inhibited from outputting an interrupt vector during an interrupt acknowledge cycle. The vector can, therefore, be provided by external hardware if necessary. It has no effect on the setting of the Interrupt Under Service bits.

INTERRUPT UNDER SERVICE VERTICAL EVENT--IUSV (D5)
This status bit is automatically set if IPV (Interrupt Pending Vertical Event) is the highest priority interrupt request pending when an Interrupt Acknowledge sequence takes place. It can also be set or cleared directly by CPU command. While the IUSV is set, internal and external daisy-chains prevent the same and lower priority sources of interrupt from requesting interrupts. The IUSV can be cleared to "Ø" only by CPU command. For details of interrupt operation see Section 2.7.

INTERRUPT ENABLE VERTICAL EVENT--IEV (D4)
This bit enables or disables the vertical event interrupt logic.

IEV=Ø: The Vertical Interrupt is disabled. The CRTC does not request an interrupt at vertical event nor responds to an interrupt acknowledge.

IEV=1. The Vertical Interrupt is enabled.
Interrupt Enable (IEV) does not affect the normal operation of Interrupt Pending (IPV) and Interrupt Under Service (IUVS). If IEV disables the interrupt (IEV=Ø), then setting the Interrupt Pending Bit (IPV) does not activate the Interrupt Request Line. If IEV=/, then a "1" in IUSV will affect the interrupt daisy chain; all lower priority devices are disabled.

INTERRUPT PENDING VERTICAL EVENT--IPV (D3)
IPV is a status bit which, when set to "1", indicates that a vertical event has occurred and needs CPU notification. A vertical event occurs when the CRTC internal load row counter matches the VERTINT value loaded in the HSYNC/VERTINT register. This interrupt provides real time positional information. This is the lowest priority IP bit inside the CRTC. The IPV can be cleared only by a CPU command.

INTERRUPT UNDER SERVICE SOFT-SCROLL--IUSS (D2)
Same as vertical event but applies for soft-scroll event.

INTERRUPT ENABLE SOFT-SCROLL--IES (D1)
This bit enables or disables the soft-scroll's interrupt logic. Same as vertical event.

INTERRUPT PENDING SOFT-SCROLL--IPS (DO)
IPS is a status bit which when set, indicates that a soft-scroll event requires CPU intervention. This is the highest priority IP bit.

ATTRIBUTE PORT ENABLE REGISTER:

Bits DØ through D1Ø in the Attribute Port Enable Register allow the corresponding attribute information to be output on the matching attribute pin (Figure 2.9). When reset, the corresponding attribute pin will be driven Low. When set, the corresponding pin will output attribute information. Bits D3 and D4 of this word affect the subscript and superscript attribute pin operation. If these bits are enabled for subscript or superscript, the corresponding pins will be active. These attributes are independent of the RØ-R4 outputs. The user can thus address a separate character font ROM for subscript or superscript display. The CURSOR PIN ENABLE (CPE, D13) bit of this register enables/disables only the cursor pin. When disabled, neither the X-Y cursor nor the attribute cursor is output through the cursor pin (CURSOR=Low).

ATTRIBUTE CURSOR ENABLE (ACE, D14)
The Attribute Cursor Enable Register enables/disables the path between attribute cursor in formation and cursor pin output.

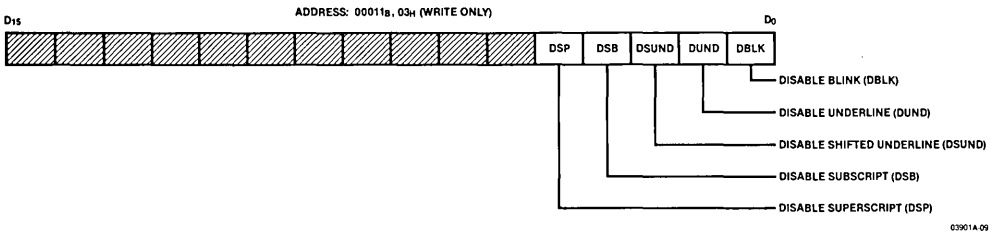


Figure 2.10. Attribute Redefinition Register

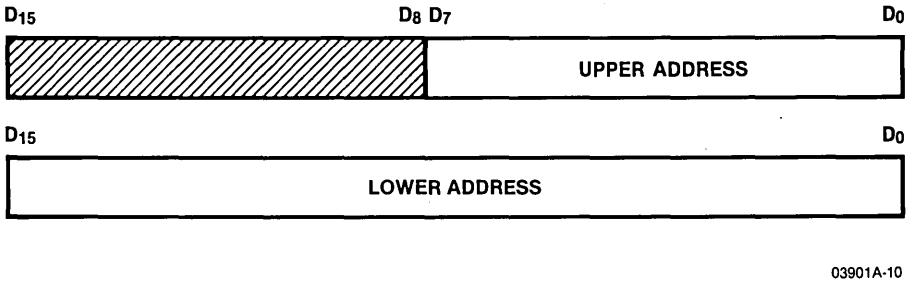


Figure 2.11. Top of Page and Top of Window Pointer Formats with L/$\overline{S}$ = 1

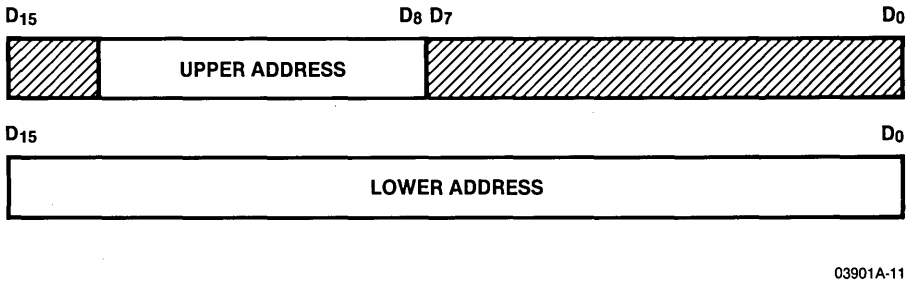


Figure 2.12. Top of Page and Top of Window Pointer Formats with L/$\overline{S}$ = 0

ATTRIBUTE REDEFINITION REGISTER:

The Attribute Redefinition Register allows the user to redefine some of the internally processed attributes, which can, therefore, be treated as user definables (Figure 2.10). A "Ø" keeps normal attribute operation, a "1" directly outputs the attribute state to its corresponding pin without any internal processing of the attributes.

TOP OF PAGE/TOP OF WINDOW REGISTERS:

Figures 2.11 and 2.12 show the format of these registers.

REGISTER	# OF ACTIVE BITS		ADDRESS		TYPE
	LINEAR	SEG.	BINARY	HEX	
TOP OF PAGE SOFT (HI)	8	7	0 0 1 0 0	Ø4	R/$\overline{W}$
TOP OF PAGE SOFT (LO)	16	16	0 0 1 0 1	Ø5	R/$\overline{W}$
TOP OF WINDOW SOFT (HI)	8	7	0 0 1 1 0	Ø6	R/$\overline{W}$
TOP OF WINDOW SOFT (LO)	16	16	0 0 1 1 1	Ø7	R/$\overline{W}$
TOP OF PAGE HARD (HI)	8	7	0 1 0 0 1	Ø9	R/$\overline{W}$
TOP OF PAGE HARD (LO)	16	16	0 1 0 1 0	ØA	R/$\overline{W}$
TOP OF WINDOW HARD (HI)	8	7	0 1 0 1 1	ØB	R/$\overline{W}$
TOP OF WINDOW HARD (LO)	16	16	0 1 1 0 0	ØC	R/$\overline{W}$

The Top Of Page and Top Of Window Registers point to the Main Definition Block and Window Definition Block respectively; these blocks contain the primary information concerning the background display and the window display.

Two different forms of top of page/window register writes are available, hard and soft. Top of page/window soft is used to trigger the soft-scroll and to interact with the soft-scroll controller (see section on soft-scroll). Top of page/window hard has no effect on the soft-scroll procedure, and should be used for link manipulations that do not involve soft scroll. If the Top of Window Register contains "Ø", no window is displayed on the screen.

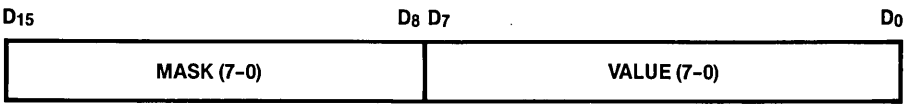
ATTRIBUTE FLAG REGISTER:

The attribute flag register defines the bit pattern that will invoke an attribute word from the attribute segment (Figure 2.13).

This 16-bit register is devided into two sections MASK and VALUE.

D15-D8 MASK (7-Ø)
D7-DØ VALUE (7-Ø)

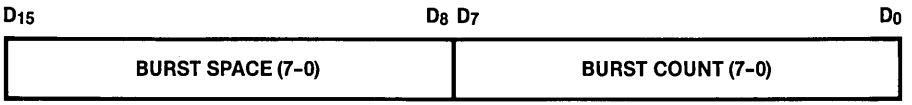
ADDRESS: 01000_B, 08_H (WRITE ONLY)



03901A-12

Figure 2.13. Attribute Flag Register

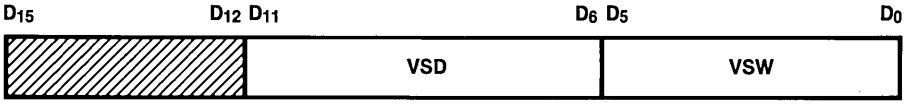
ADDRESS: 10000_B, 10_H (WRITE ONLY)



03901A-13

Figure 2.14. Burst Register

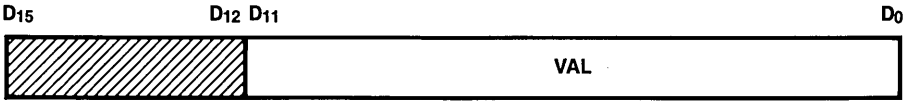
ADDRESS: 10001_B, 11_H (WRITE ONLY)



03901A-14

Figure 2.15. Vertical Sync Width/Vertical Scan Delay Register

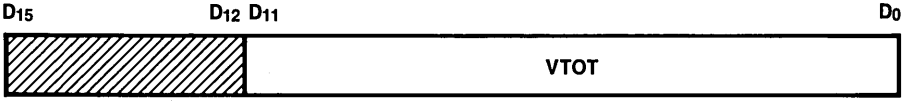
ADDRESS: 10010_B, 12_H (WRITE ONLY)



03901A-15

Figure 2.16. Vertical Active Lines Register

ADDRESS: 10011_B, 13_H (WRITE ONLY)



03901A-16

Figure 2.17. Vertical Total Lines Register

MASK (7-Ø) (D15-D8)
The mask register defines which bits of the 8-bit character field will be compared against the value register to determine if the character invokes an attribute word. A Ø in bit position N of the mask indicates that character bit N is a "don't care" in the value comparison. A "1" in bit position N of the mask indicates that character bit N should be compared against value bit N.

VALUE (7-Ø) (D7-DØ)
The value register holds up to 8 bits of information for comparison with the fetched character, to determine if an attribute is invoked. Note that only those bits of the value register which have the corresponding bits of the mask register set to "1" are compared against the character.

Example 1: We want all the control characters (characters between $ØØ_H$ and $1F_H$) to invoke an attribute. These characters are of the form:

	Ø Ø 0 X X X X X (X IS DON'T CARE)
THE MASK WILL BE	1 1 1 0 0 0 0 0
THE VALUE WILL BE	0 0 0 X X X X X

so the attribute flag register contents are:

1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 ($EØØØ_H$)

Example 2: We want one specific flag ($7F_H$) to invoke an attribute:

In this case we want to compare all the bits of the character to the value, so the mask is:

	1 1 1 1 1 1 1 1
and the value is:	0 1 1 1 1 1 1 1 ($7F_H$)

Hence the attribute register contains

1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 ($FF7F_H$)

If the user wishes to disable the attribute fetch mechanism a practical approach is: Set all bits in the MASK register and load VALUE with an unused character code.

BURST REGISTER:

The Burst Register (Figure 2.14) specifies the bus occupancy of the CRTC DMA unit. BURST COUNT determines the maximum burst length. BURST SPACE determines the minimum release time between two bursts. This guarantees realtime responses of the CPU or other peripherals. BURST COUNT and BURST SPACE must be programmed with values allowing the CRTC to fetch all data needed for a flicker-free screen.

D15-D8	BURST SPACE (BS7-Ø)
D7-DØ	BURST COUNT (BC7-Ø)

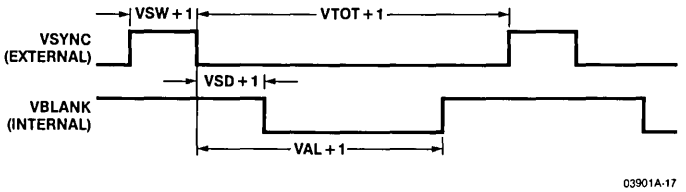


Figure 2.18. Non-Interlaced Video Vertical Sync Timing

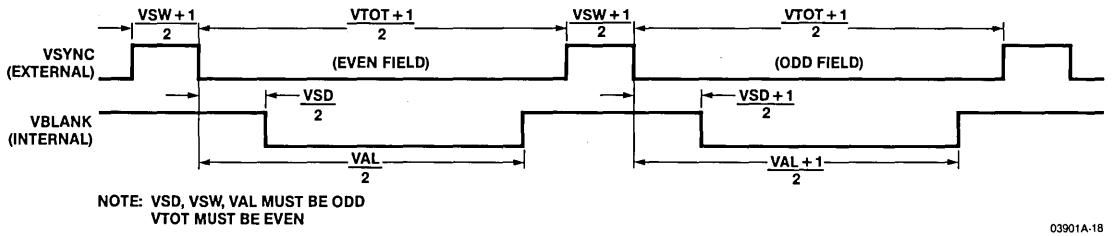


Figure 2.19. RFI and Video Interlace Vertical Sync Timing

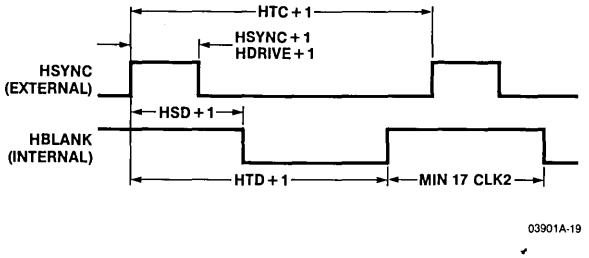


Figure 2.20. Horizontal Sync Timing

BURST SPACE--BS7-Ø (D15-D8)
This 8-bit count specifies the number of 15 system clock cycle (CLK1) periods before another bus request will be issued. This count is triggered at the end of a DMA activity. If the count is set to "0" the CRTC occupies the bus as long as necessary to accomplish its DMA activity.

BURST COUNT---BC7-Ø (D8-DØ)
This 8-bit count specifies the number of word data transfers during DMA operation. Each count is equivalent to one DMA transfer cycle. If BS(7-Ø) is set to "Ø", no DMA activity will occur.

VIDEO TIMING REGISTERS:

These registers are initialized before setting the DE bit in Mode Register 1. They hold the parameters needed to generate vertical and horizontal sync and blank (VSYNC, HSYNC, and BLANK). These signals are put out on the like-named pins of the CRTC and are used by the Am8152A/53A. BLANK is a composite of horizontal and vertical blank (HBLANK and VBLANK).

Horizontal timing parameters are expressed in number of system or character clock cycles (CLK1/2 bit of Mode Register 1). Vertical timing parameters are expressed in number of scan lines.

HSYNC (8 bit counter) and HDRIVE (9 bit counter) represent two ways of specifying the signal waveform on the HSYNC output pin. HDRIVE usually determines a signal of 50% duty cycle because some monitors require a close to 50% duty cycle for the horizontal frequency.

In the following discussion a frame consists of one field in non-interlaced mode and two fields (even and odd) in RFI and video interlace modes. Figures 2.18 and 2.19 show the vertical timing.

VERTICAL SYNC WIDTH/VERTICAL SCAN DELAY REGISTER:

Figure 2.15 shows the register format.

D15-D12	NOT USED
D11-D6	VERTICAL SCAN DELAY (VSD)
D5-DØ	VERTICAL SYNC WIDTH (VSW)

VERTICAL SCAN DELAY--VSD (D11-D6)
The vertical scan delay field specifies the vertical blank time after the falling edge of VSYNC, thus defining the top border width, or vertical back porch of the screen. VSD is expressed in scan-line units. When in non-interlaced mode, the actual vertical scan delay is equal to VSD + 1 scan lines. When in video interlace mode or RFI (Repeat Field Interlace) mode, the actual vertical scan delay is equal to (VSD + 1)/2 lines. In this case VSD must be odd.

VERTICAL SYNC WIDTH--VSW (D5-DO)
The vertical sync width determines the width of the active High pulse signal which is sent through VSYNC output to the CRT monitor in order to synchronize it vertically.

VSW is expressed in scan line units. In non-interlaced mode, the actual vertical sync width is equal to VSW + 1 scan lines.

In interlaced and RFI mode, the actual vertical sync width is equal to (VSW + 1)/2 lines. In this case VSW must be odd.

VERTICAL ACTIVE LINES REGISTER:

D15-D12 NOT USED
D11-DØ VERTICAL ACTIVE LINES (VAL)

This 12-bit field defines the number of scan lines between the end of a vertical sync pulse and the start of vertical blanking (Figure 2.16).

When in non-interlaced mode, the actual scan-line number between the falling edge of VSYNC and the rising edge of VBLANK is equal to VAL + 1. The active video area height on the screen is then (VAL + 1) - (VSD + 1)=VAL - VSD scan lines.

When in video interlace or RFI mode, the actual scan-line number between VSYNC and VBLANK is equal to (VAL + 1)/2. In this case VAL must be odd. The active video area height on the screen is then given by (VAL + 1)/2 - (VSD + 1)/2 = (VAL - VSD)/2 scan lines. This is true for the odd and even field.

VERTICAL TOTAL LINES REGISTER:

D15-D12 NOT USED
D11-DØ VERTICAL TOTAL LINES (VTOT)

The vertical total lines register defines the total number of scan lines per field minus the vertical sync width (Figure 2.17).

In non-interlaced mode, the actual scan line number between VSYNC and next VSYNC is (VTOT + 1).

In interlaced or RFI mode, this timing is (VTOT + 1)/2, and VTOT must be even (half scan line between even and odd fields).

Figure 2.20 shows the horizontal timing.

HORIZONTAL SYNC AND VERTICAL INTERRUPT ROW REGISTER:
Figure 2.21 shows the register format.

D15-D8 VERTICAL INTERRUPT ROW (VERTINT)
D7-DØ HORIZONTAL SYNC WIDTH (HSYNC)

VERTICAL INTERRUPT ROW--VERTINT (D8-D15)
This field determines the row number which, after being completely loaded by DMA, causes an interrupt.

HORIZONTAL SYNC WIDTH--HSYNC (DØ-D7)
This field determines the width of the horizontal sync (active High) pulse in terms of video clock units (CLK1 or CLK2 depending upon CLK1/2 bit in Mode Register 1), provided that HSYNC is selected (HOS=Ø in Mode Register 1). These pulses are output on the HSYNC pin. The actual width of the signal is HSYNC + 1 clock periods.

HORIZONTAL DRIVE REGISTER:

D15-D9 NOT USED
D8-DØ HORIZONTAL DRIVE (HDRV)

This register determines the interval from HSYNC to HSYNC if horizontal drive is selected (HOS=1 in Mode Register 1). The interval is HDRV + 1 clock periods. This is also an output on the HSYNC pin. (See Figure 2.22).

HORIZONTAL SCAN DELAY REGISTER:

D15-D9 NOT USED
D8-DØ HORIZONTAL SCAN DELAY (HSD)

The horizontal scan delay register determines the interval from rising edge of HSYNC to the falling edge of HBLANK, which defines the left border on the screen. The actual interval value is (HSD + 1) clock periods (See Figure 2.23).

HORIZONTAL TOTAL COUNT REGISTER:

D15-D1Ø NOT USED
D9-DØ HORIZONTAL TOTAL COUNT (HTC)

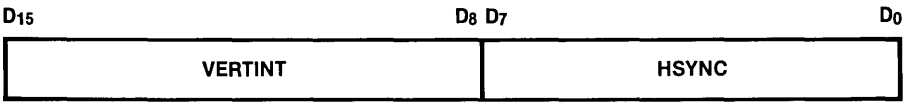
This register determines the period of the HSYNC waveform. The period is (HTC + 1) clock periods (See Figure 2.24).

HORIZONTAL TOTAL DISPLAY REGISTER:

D15-D10 NOT USED
D9-D0 HORIZONTAL TOTAL DISPLAY (HTD)

This register determines the interval from the rising edge of HSYNC to the rising edge of HBLANK. HTD must be odd in interlaced mode. The actual interval value is (HTD+1) clock periods (See Figure 2.25).

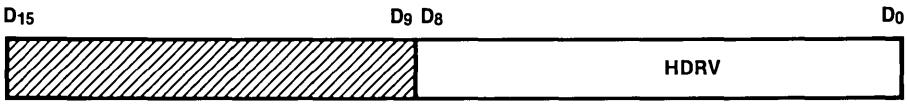
ADDRESS: 10100_B, 14_H (WRITE ONLY)



03901A-20

Figure 2.21. Horizontal Sync Width/Vertical Interrupt Row Register

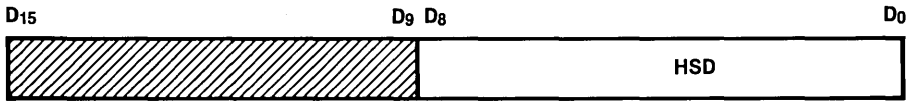
ADDRESS: 10101_B, 15_H (WRITE ONLY)



03901A-21

Figure 2.22. Horizontal Drive Register

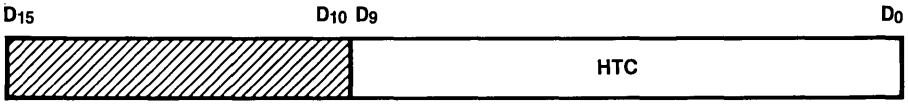
ADDRESS: 10110_B, 16_H (WRITE ONLY)



03901A-22

Figure 2.23. Horizontal Scan Delay Register

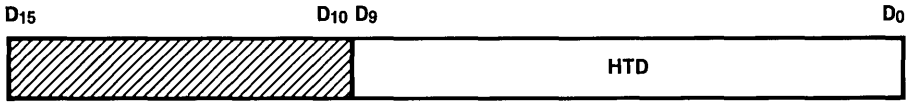
ADDRESS: 10111_B, 17_H (WRITE ONLY)



03901A-23

Figure 2.24. Horizontal Total Count Register

ADDRESS: 11000_B, 18_H (WRITE ONLY)



03901A-24

Figure 2.25. Horizontal Total Display Register

2.3.4 Video Timing Programming Example

We want to build a 30 row by 80 character display, each character embedded in a 8 x 17 (HxV) matrix, with a refresh rate of 50 Hz in non-interlaced mode using a CRT monitor with the following characteristics:

Pixel Resolution:	1280 pixels horizontal 1024 pixels vertical
Scanning frequency:	28 - 36 kHz horizontal 45 - 65 Hz vertical
Horizontal retrace time:	6 microseconds
Vertical retrace time:	600 microseconds
Horzontal SYNC width:	3 microseconds

Solution:
The appropriate character clock and the timing parameters for the video timing register have to be calculated.

The active display size is given by:
Horizontal: 80 characters x 8 pixels/char=640 pixels
Vertical: 30 rows x 17 scan lines/row=510 scan lines

Assuming a 20% blank border vertically, the 510 scan lines occupy 80% of frame time. At a frame rate of 50Hz, the horizontal frequency can be calculated as:
Total Scan Lines/frame: 510 scan lines/0.80=637 scan lines
Horizontal Frequency: 637 x 50 Hz=31.85 kHz

Assuming a 20% blank horizontally, the 80 characters occupy 80% of row time. Character clock is therefore 100 times the horizontal frequency. Each character occupies 1/100 of the row.

Let us use a more convenient frequency, 3.00 MHz, as character clock and re-calculate the parameters:

Character clock	3.00 MHz
Horizontal frequency	30kHz
Scan line time	33.3 microseconds
Frame time	637x33.3 microsec = 21.2 ms
Frame rate	47 Hz

Now the registers' contents can be calculated:

Mode Register 1:

The character clock is 3 MHz; the CLK1/2 bit is set to "Ø" to select CLK2 for the frame timing generation.

With only 80 characters/row, we select "SLIM=1" which reduces the row buffer length to 96 characters.

The monitor accepts an HSYNC signal: "HOS=Ø"

Non-interlaced mode yields in: "I1=Ø", "IO=Ø".

External Sync EnableØ since we do not need to be synchronized to another signals.

Display Enable should be set to "1", once the other registers are set to the proper values.

Vertical Active Line register:

This value is the total scan line number of the screen minus the number of scan lines contained in the bottom border area (10% of the screen height):

VAL + 1=0.9 x (VTOT+1) - 0.9x619 - 557 scan lines
VAL = 556_{10} = $22C_H$

ADDRESS: 10010_B (12_H) VAL = $022C_H$

Vertical Total Line register:

VTOT + 1 = 637-(VSW+1) - 619
VTOT = 618_{10} = $26A_H$

ADDRESS: 10011_B (13_H) VTOT = $026A_H$

Vertical sync width/vertical scan delay register:

Vertical sync width:
VSW + 1 = 600 usec
VSW+1 = 600/33.3 = 18 scan lines
VSW = 17_{10} = 11_H

Vertical scan delay: (the top border width of the screen)
VSD + 1 = (VAL + 1) - 510
VSD = 46_{10} = $2E_H$
VSD shifted 6 bits left to fit the field in the register.
VSD_{shift} = $B80_H$

ADDRESS: 10001_B (11_H) WSD/VSW = $0B91_H$

Horizontal sync and vertical interrupt row register:

We will set VERTINT to 0 in this example.
HSYNC + 1 = 3 usec = 3 x 3 = 9 character-clocks.
HSYNC = 8_{10} = 8_H

ADDRESS: 10100_B (14_H) VERTINT/HSYNC = 0008_H

Horizontal drive register:

Is don't care since HOS = 0. (HSYNC selected)

Horizontal scan delay register:

(the left border area of the screen)
HSD + 1 = (100-9-80)/2+9 = 15 character clocks
HSD = 14 = $0E_H$

ADDRESS: 10110_B (15_H) HSD = $000E_H$

Horizontal total count register:

HTC + 1 = 100 character clocks
HTC = 99_{10} = 63_H

ADDRESS: 10111_B (17_H) HTC = 0063_H

Horizontal total display register:

HTD + 1=80+15
HTD = 94_{10} = $5E_H$

ADDRESS: 11000_B (18_H) HTD = $005E_H$

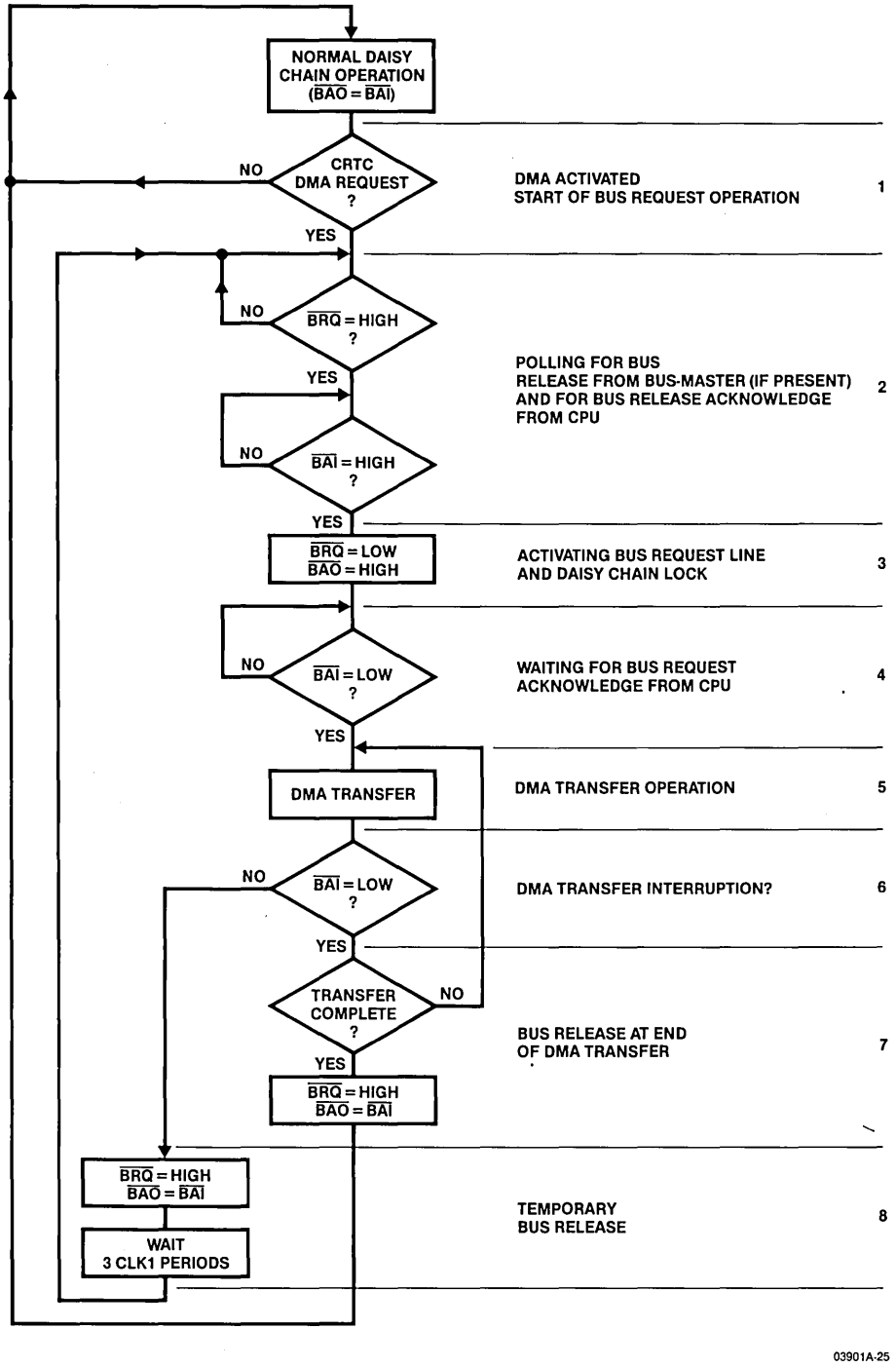


Figure 2.26. DMA Bus Request Flow Chart

2.4 DMA OPERATION

Once the CRTC has been initialized, and the various registers have been programmed to meet the application's needs, the CRTC is responsible for initiating system bus requests in order to fetch control data and display data from memory, and transfer them into its on-board-registers and row buffers respectively. The CRTC tries to get the bus after the DE bit in Mode Register 1 has been set to a "1".

2.4.1 DMA Signals And Protocol:

Before the CRTC can perform a DMA operation, it must gain control of the system bus. The $\overline{BRQ}$, $\overline{BAI}$ and $\overline{BAO}$ interface pins constitute the basic interface between the CRTC and other devices capable of bus preemption (DMA devices, microprocessors, etc.) to arbitrate which device has control over the system bus. Whenever the CRTC wants to gain bus control, the operation is executed according to the flowchart in Figure 2.26.

DMA SEQUENCE DESCRIPTION:

1. If the CRTC needs to perform a DMA access, it triggers the bus request operation.

2. First it checks whether the bus is mastered by another peripheral device by polling the $\overline{BRQ}$ line until it is High. Then it waits for the acknowledgement of bus release from the CPU through the daisy-chain ($\overline{BAI}$ = High).

3. At that time the bus is under control of the CPU, and the CRTC can issue its request by pulling $\overline{BRQ}$ Low. It also inhibits Bus Acknowledge to ripple toward lower priority devices (in the lower part of the daisy-chain) by pulling $\overline{BAO}$ High; this avoids bus preemption from a lower priority device which issued the $\overline{BRQ}$ at the same time the CRTC did.

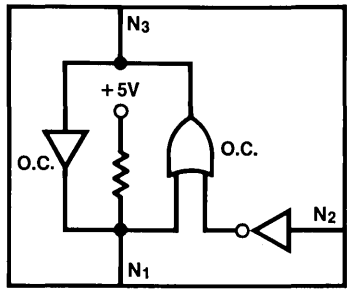
4. Before initiating any DMA transfer, the CRTC waits for bus request acknowledge from the CPU by polling its $\overline{BAI}$ input.

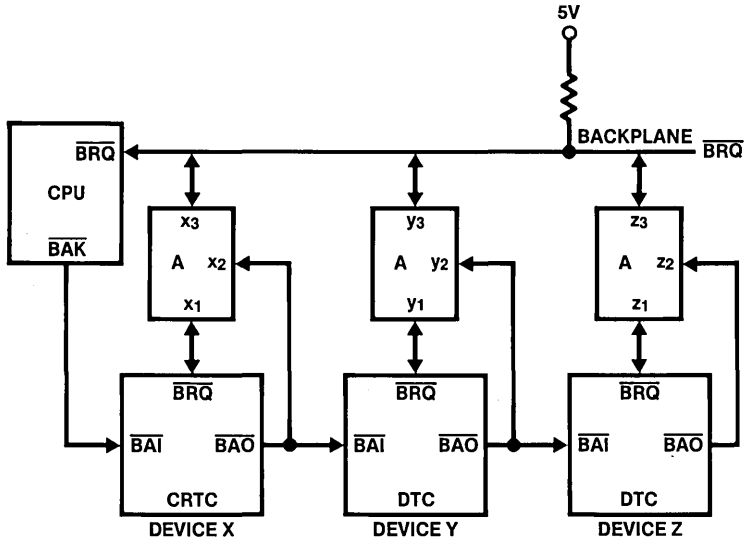
5. DMA transfers are processed here, the CRTC now acts as the bus master.

6. The CRTC DMA transfer can be interrupted by removing Bus Acknowledge In ($\overline{BAI}$ = High).

7. The CRTC terminates transfer when it has filled the internal row buffers or when the burst count is timed out. The bus is released ($\overline{BRQ}$ = High) and Bus Acknowledge ripples through ($\overline{BAO}$ = $\overline{BAI}$). Then either the CPU or a lower priority device on the daisy chain can gain control over the bus. The lower priority device might have pulled $\overline{BRQ}$ Low concurrently to the CRTC and is waiting for $\overline{BAI}$ = Low to start its activity.

DETAIL "A"





03901A-26

Figure 2.27. System with Multiple DMA Devices

8. The CRTC DMA transfer is interrupted by removing $\overline{BAI}$. The CRTC finishes the current bus cycle and releases the bus for three system clocks ($\overline{BRQ}$ = High, $\overline{BAO}$ = $\overline{BAI}$). Then it tries to resume DMA activity and continues DMA operations and burst count from where it was interrupted.

2.4.2 Buffering $\overline{BRQ}$

When $\overline{BRQ}$ needs to be buffered, e.g. to drive a system backplane, then a specific bidirectional interface buffer must be used. Such an interface and its implementation is described below:

Detail "A" in Figure 2.27 shows the $\overline{BRQ}$ buffer logic. Note that the "buffer" and the "OR gate" are both open collector (O.C.) devices. When a High is on the backplane $\overline{BRQ}$, and no DMA device requested the bus, then all $\overline{BAI}$'s and $\overline{BAO}$'s are High, hence X3 and X2 are High and X1 is driven High.

If device X requests the bus, it will lock $\overline{BAO}$ High and pull X1 Low to initiate a bus request, which will in turn pull X3 Low since X2 is High ($\overline{BAO}$ High). The detail "A" logic is then locked into this state through the open collector buffer, as long as X2 remains High. The request is also transmitted to the CPU and the other detail "A" interfaces on the bus. All these interfaces will be locked the same way as the requesting one. A few cycles later, the CPU acknowledges the bus request by pulling $\overline{BUSAK}$ Low, the CRTC (device X) will then execute its transfers. When the CRTC finishes its transfers, it releases $\overline{BRQ}$ and relinks its $\overline{BAI}$ input to $\overline{BAO}$ output, hence driving $\overline{BAO}$ Low. The Low will propagate through the daisy chain, and as long as one of the $\overline{BAO}$ is High, the backplane $\overline{BRQ}$ line and the devices $\overline{BRQ}$ signals will be held Low due to detail "A" logic structure.

Once all the $\overline{BAO}$'s have gone High, the backplane $\overline{BRQ}$ goes High, and the CPU gains control over the bus.

2.4.3 DMA Transfer Operation

The DMA transfer itself consists of data moves from memory into the CRTC, controlled by the CRTC's DMA unit.

If a control block is fetched, the words loaded will be steered toward the internal control registers. If display data (characters or attributes) is fetched from memory, it will be steered toward an internal row buffer.

In both cases the CRTC will have to:
1. Output the address of the data location
2. Sample the $\overline{WAIT}$ input and stretch the read cycle if needed. $\overline{WAIT}$ is sampled only at the falling edge of the system clock in T2.
3. Read the data and transfer it to the proper destination (buffer or internal register)

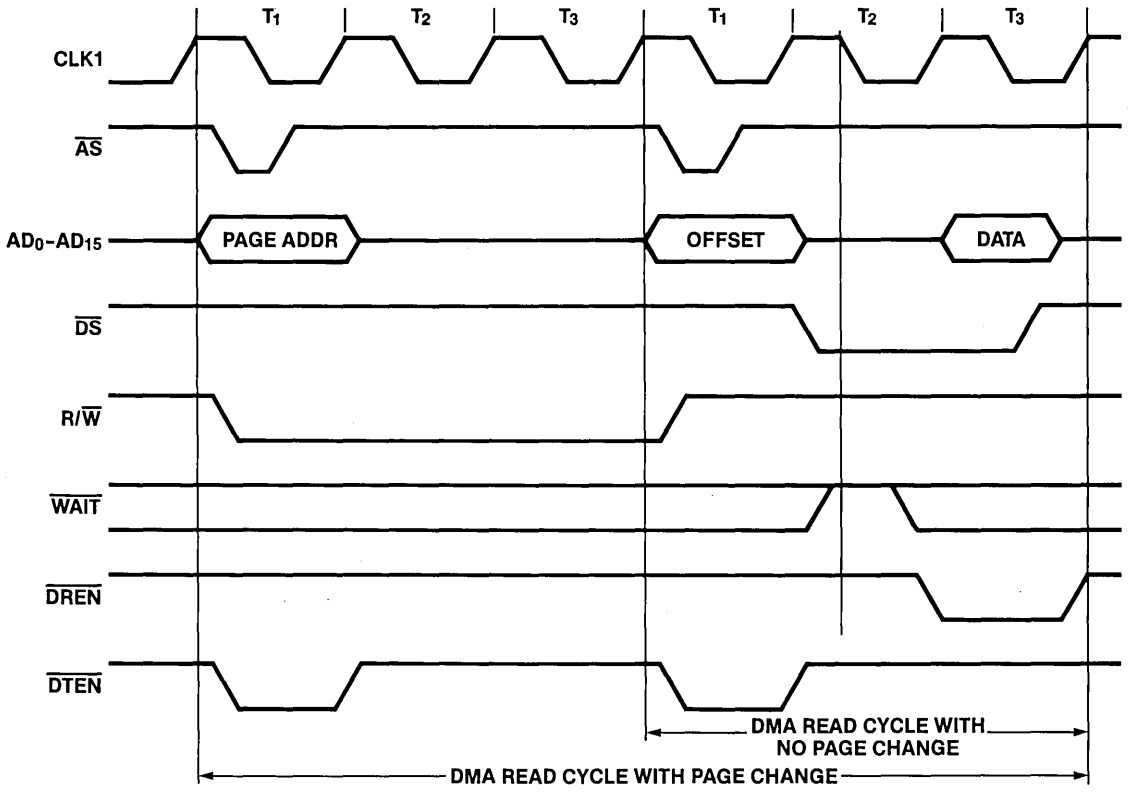


Figure 2.28. DMA Transfer Operation

Since the Am8052 can address up to 16 Mbytes addressed as 256 pages of 64Kbytes each, there are two different methods for fetching data:

1. There is a page change between the previous fetch cycle and the current one, or this is the first fetch of a new frame. In either case, succeeding read cycles will be preceeded by a single write cycle to latch the new page address. (See Figure 2.28.)
2. There is no page change since the previous fetch cycle and it is not the first fetch of a new frame. In this case the succeeding fetches will not be preceeded by a page address write cycle. A new burst does not necessarily begin with a page address update.

DMA READ AND WRITE OPERATIONS:
The start of a DMA cycle is initiated by $\overline{AS}$ being driven Low, which indicates a valid address on the AD0-AD15 address-data lines. At that time $\overline{DTEN}$ is also driven Low and allows the valid address to be buffered on the system bus through external buffers. The valid address may be latched on the system bus on the rising edge of $\overline{AS}$.

During the first portion of a DMA read cycle with page change, R/$\overline{W}$ is pulled Low by the CRTC for three complete clock cycles, and the address present on the AD0-AD7 bus during T1 is the updated page address which should be latched externally on the rising edge of $\overline{AS}$. The CRTC never outputs an active $\overline{DS}$ during a write cycle. The next three clock cycles represent a normal DMA read cycle.

During T2 of these 3 clock cycles, the CRTC ceases driving the AD0-AD15 bus with the address information, and $\overline{DTEN}$ goes High. $\overline{DS}$ is driven Low as an indication to the memory system that it may drive the bus with the read data. Half of a clock cycle later, $\overline{DREN}$ is driven Low to enable the receiving buffers local to the CRTC.

Data is captured by the CRTC on the falling edge of the T3 clock cycle; then both $\overline{DS}$ and $\overline{DREN}$ return High.

WAIT OPERATION:
During T2 of the read cycle, the $\overline{WAIT}$ signal is sampled by the falling edge of CLK1. If Low, the cycle will be stretched by one CLK1 cycle.

The CTRC also has a software Wait state capability: 0, 1 or 2 Wait states can be specified in Mode Register 1 and are automatically inserted in each read cycle independently of the $\overline{WAIT}$ input line.

When both hardware and software Wait states are requested, they occur consecutively and not concurrently: The hardware Wait states is honored first, immediately followed by software Wait states if so programmed.

DUMMY DMA CYCLES:
A dummy DMA cycle takes place when a window row is filled with the fill code. At that time, the DMA keeps the bus mastership, although it does not fetch any character on this bus since the fill code is stored internally. Dummy DMA cycles are also executed under some circumstances when doing internal processing. Each dummy DMA cycles takes three system clock (CLK1) cycles to execute.

2.4.4 DMA Burst Control

During DMA action, the CPU is denied access to the bus and therefore cannot execute programs. This situation can lead to problems in the interrupt response time of the CPU, since the CPU can only recognize and service an interrupt request while in control of the bus. Note that at the beginning of every frame, immediately after the vertical blanking interval goes active, the CRTC tries to request the bus.

To allow the CPU control of the bus within certain limits, a burst register is provided inside the CRTC and is programmable by the CPU. This burst register specifies a time slot during which the CRTC is allowed to request the bus. Both the time slot duration and its cycle time are programmable. For further information refer to the register section.

2.5 ROW MANAGEMENT UNIT OPERATION

2.5.1 Introduction

The Row Management Unit is in charge of the system for fetching, interpretation and steering the information contained in memory, to load the three row-buffers with displayable information and to update internal registers to redefine some of the screen characteristics.

Listed below is the information that the Row Management Unit may steer for updating.

Steer into the row-buffers:
- characters
- attributes

Steer into the internal registers:
1. alterable on a frame basis:
- absolute cursor coordinates (CUX, CUY)
- fill character code
- blink control and parameters (for cursors and characters)
- scroll control and parameters
- interrupt vectors (for vertical event and softscroll event)

2. alterable on a row basis:
- total scan line count per row (TSLC)
- normal character start and end line numbers (NCS, NCE)
- superscript character start and end scan-line numbers (SPCS, SPCE)
- subscript character start and end scan-line numbers (SBCS, SBCE)
- cursor pattern start and end scan-line numbers (CURS, CURE)
- underline position (UND)
- shifted underline position (SUND)

The information to be fetched by the Row Management Unit is addressed by linked-list pointers, and the Row Management Unit keeps track of the addresses of the information present in memory.

The Row Management Unit also interprets window information when it is present.

The final task performed by the Row Management Unit is the selection of displayable characters (which are the only ones loaded into the row buffers) depending upon the "ignore" and "invisible attribute flag" bits settings.

2.5.2 Data Structure

The CRTC is capable of controlling and displaying a text file on the screen (known as background) concurrently with other text files embedded in rectangles (known as windows) positioned anywhere inside the active display area of the screen. With conventional CRT controllers, this feature can only be implemented if the CPU is aware of the position and size of the window, with all the inconvenience and software complexity this implies. One of the important features of the Am8052 CRTC is that it allows the CPU to process a background file and a window file independently without continuously being concerned with size and position of the window.

The CRTC holds two pointer registers; each containing the starting address of a linked-list residing in memory: one pointer corresponds to the background information, while the other corresponds to the first window's information. The first window is the first one encountered when scanning the screen from top to bottom. Note that the user is able to define an arbitrary number of windows on the screen, as long as two background character rows (three for interlaced video) separate the windows vertically. Virtual windows, however, may occur side by side (horizontal split screen).

Each window links to the following one (ranging from top to bottom of the screen) with a link pointer. There are no more windows when the link pointer of the last window contains zero.

So far, we have explained that the CRTC is aware of two main lists in memory:

1. The background list pointed to by top of page (TOP) register, containing the parameters of the background display.

2. The window(s) list pointed to by top of window (TOW) register, containing the parameters of the window(s) display.

Depending upon the memory addressing scheme, the user can choose either of two addressing modes: segmented more or linear mode:

SEGMENTED MODE:
The segmented mode divides the memory into pages containing 64K bytes each. The CRTC can address 128 pages. In this case, the pointer is 23 bits wide arranged in two 16-bit words with the following configuration:

- 7 bits pointing to one page among the 128 addressable pages. These 7 bits are right justified in the most significant byte of first 16-bit word.

- 16 bits pointing to the address within the selected page. These 16 bits constitute the second word.

When operating in the segmented mode, crossing a page boundary does not increment the page number. It results in wrap-around operation within the same page.

LINEAR MODE:
In the linear mode the CRTC addresses memory as one 16 megaword block, with a 24-bit wide pointer arranged in two 16-bit words with the following configurations:

- 8 bits representing the most significant part of the address embedded in the least significant byte of the first word.

- 16 bits representing the least significant part of the address in the second word.

In this mode, when the second word crosses a 64K boundary, the first word is incremented by one.

The selection between these two modes is accomplished through the $L/\overline{S}$ bit in Mode Register 1:

$L/\overline{S}$ = Ø	segmented mode enabled
$L/\overline{S}$ = 1	linear mode enabled

Consistent with the byte addressing method used by all 16-bit microprocessors, ADØ will always output a "Ø" at address time. This means that the CRTC actually addresses 32K 16-bit words instead of 64K bytes. This applies for both linear and segmented addressing modes.

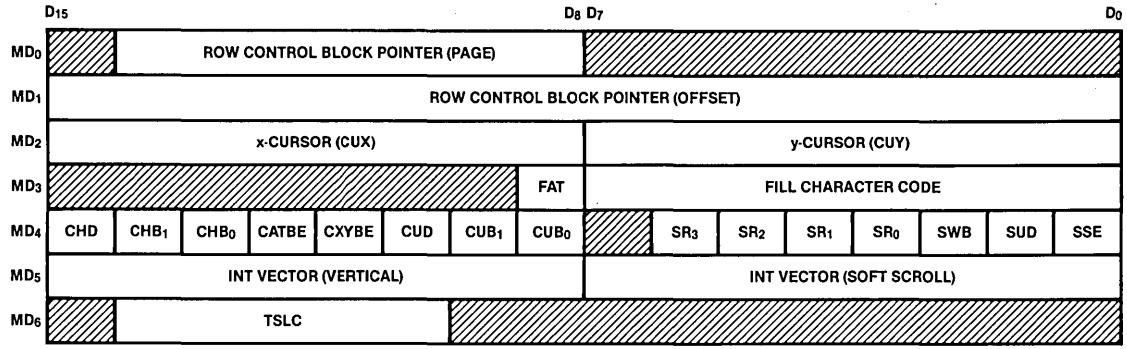
Let us see now how background and window information is organized. We will first focus on the background list.

2.5.3 Background Information Management

The "TOP" register points to the first data word of a block called "Main Definition Block". This block is unique for each background list, and the information it contains will be fetched on a frame basis and stored into the applicable internal registers of the CRTC.

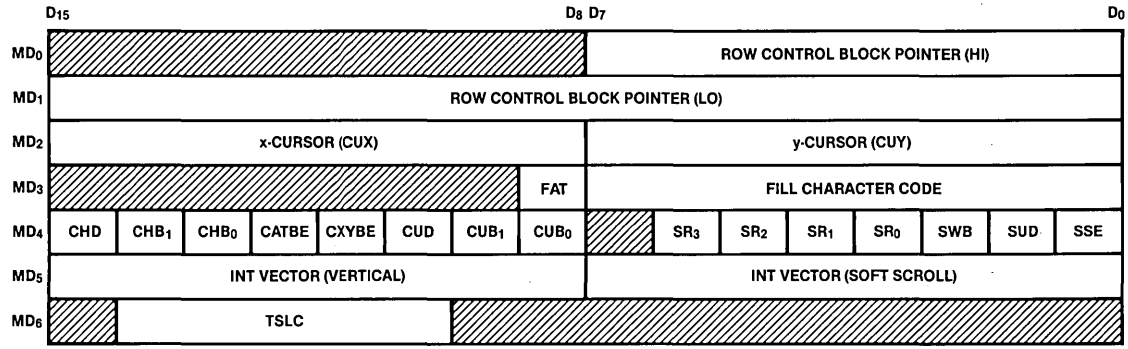
MAIN DEFINITION BLOCK (MDB) OVERVIEW:
The Main Definition Block contains seven data words (MØ-6) defined as follows (Figure 2.29 and 2.30):

MDØ,MD1	-	"Row Control Block" pointer
MD2	-	Absolute cursor coordinates ("X" coordinate byte and "Y" coordinate byte)



03901A-28

Figure 2.29. Main Definition Block ($L/\overline{S} = 0$)



03901A-29

Figure 2.30. Main Definition Block ($L/\overline{S} = 1$)

MD3 - Fill character code (1 flag bit + 1 byte code)
MD4 - Blink control/scroll control
MD5 - Interrupt vectors: vertical event/scroll event
MD6 - Total scan line count per row

MDB DETAILED DESCRIPTION:

MDØ,MD1 - The Row Control Block pointer points to the block defining the first row's control information.

MD2 - The absolute cursor coordinates indicates the row number and the character position within this row where the absolute cursor is displayed. The topmost row is row "Ø" the leftmost character position is "Ø".

MD3 - The fill character code is a user-defined 8-bit code. This is used as a filler in the row buffer if all the characters for that row have been loaded and did not fill the programmed buffer size. Segments with a character code pointer of "Ø" are also filled with the fill code. The number of visible characters (visible #) specifies the length of these segments. Windows, where the window segments do not fill up the window size, are filled by the fill code too. The flag bit (flag attribute), when set, causes the CRTC to load an extra attribute word from the attribute list and use it as a latched attribute (immediately active) for the fill character. The extra attribute word must invoke a latched attribute.

MD4 - The blink control/scroll control is composed of 15 bits.

Soft Scroll Enable (SSE) enables the soft-scroll operation for either the background or a window.

Ø Soft-scroll disabled
1 Soft-scroll enabled

Scroll Up/Down (SUD) indicates the direction of the scroll.

Ø Soft-scroll down
1 Soft-scroll up

Scroll Window/Background (SWB) indicates whether the background or a window will be scrolled.

Ø Soft-scroll background
1 Soft-scroll window

Scroll Rate (SR3-Ø) is a 4-bit word specifying the soft-scroll rate according to the following table:

SR3	SR2	SR1	SR0	Scroll Rate
0	0	0	0	1 Scan Line Per Frame
0	0	0	1	2 Scan Lines Per Frame
0	0	1	0	3 Scan Lines Per Frame
0	0	1	1	4 Scan Lines Per Frame
0	1	0	0	5 Scan Lines Per Frame
0	1	0	1	6 Scan Lines Per Frame
0	1	1	0	7 Scan Lines Per Frame
0	1	1	1	8 Scan Lines Per Frame (fastest)
1	0	0	0	1 Scan Line Per 1 Frame
1	0	0	1	1 Scan Line Per 2 Frames
1	0	1	0	1 Scan Line Per 3 Frames
1	0	1	1	1 Scan Line Per 4 Frames
1	1	0	0	1 Scan Line Per 5 Frames
1	1	0	1	1 Scan Line Per 6 Frames
1	1	1	0	1 Scan Line Per 7 Frames
1	1	1	1	1 Scan Line Per 8 Frames (slowest)

Cursor Blink Rate (CUB1, CUBØ) defines the blinking rate for both attribute and absolute cursors:

CUB1	CUBØ	Blink Period	Blink Frequency (at 60Hz Frame Rate)
0	0	16 Frames	3.75 Hz
0	1	32 Frames	1.85 Hz
1	0	64 Frames	0.93 Hz
1	1	128 Frames	0.46 Hz

Cursor blink Duty cycle (CUD):

CUD	Cursor Blink Duty Cycle
Ø	Blink Output 75% Inactive, 25% Active
1	Blink Output 50% Inactive, 50% Active

Character blink Duty cycle (CHD):

CUD	Character Blink Duty Cycle
Ø	Blink Output 75% Inactive, 25% Active
1	Blink Output 50% Inactive, 50% Active

Absolute cursor Blink Enable (CXYBE):

Ø	Cursor Blink Disable
1	Cursor Blink Enable

Attribute Cursor Blink Enable (CATBE):

Ø	Cursor Blink Disable
1	Cursor Blink Enable

Character Blink Rate (CHB 1,0):

CHB1	CHB0	Blink Period	Blink Frequency (at 60Hz Frame Rate)
0	0	16 Frames	3.75 Hz
0	1	32 Frames	1.85 Hz
1	0	64 Frames	0.93 Hz
1	1	128 Frames	0.46 Hz

Note that the character and the cursor can have different blink rates and different duty cycles.

MD5 - The interrupt vector register contains the soft-scroll and vertical event interrupt vectors. When one of these interrupts is activated, the corresponding 8-bit vector is output on AD7-ADØ at interrupt acknowledge time, if the NV-bit in Mode Register 2 is reset.

The vertical event interrupt vector is totally user-programmable.

The soft-scroll interrupt vector is partially user-programmable: Bits Ø, 2-7 are user-definable while bit 1 reflects the state of the SIP (Scroll Interrupt Pending) bit. This feature allows the user to steer the soft-scroll interrupts into two different routines.

SIP = 1 The CRT is informing the CPU to execute a relink during scrolling operation.
SIP = Ø The CRT does not need CPU intervention but signals the CPU that the scroll operation is completed.

MD6 - TSLC is a 5-bit value defining the number of total scan lines per row minus one. This value is reprogrammable on a row basis via the Row Redefinition Block.

This TSLC must be equal to the TSLC of the first row in the linked-list.

In video interlace or RFI mode, the TSLCs of all rows displayed must be even or the TSLCs of all rows must be odd. In non-interlaced video, rows with odd and even TSLCs may be mixed. However, this is restricted when displaying windows (refer to Section 2.5.4). Figure 2.31 shows the values of the total number of scan lines for all video modes.

TOTAL NUMBER OF SCAN LINES		
TSLC	NON-INTERLACED MODE	INTERLACE OR RFI MODE
00000	1	1+1 = 2
00001	2	1+2 = 3
00010	3	2+2 = 4
00011	4	2+3 = 5
00100	5	3+3 = 6
00101	6	3+4 = 7
00110	7	4+4 = 8
00111	8	4+5 = 9
01000	9	5+5 = 10
01001	10	5+6 = 11
01010	11	6+6 = 12
01011	12	6+7 = 13
01100	13	7+7 = 14
01101	14	7+8 = 15
01110	15	8+8 = 16
01111	16	8+9 = 17
-		
-		
-		
-		
-		
11111	32	16+17 = 33

Figure 2.31: Total number of scan lines as a function of TSLC

ROW CONTROL BLOCK (RCB):
Once the CRTC has loaded the Main Definition Block into its internal registers, it will fetch the first Row Control Block (Figure 2.32 and 2.33). To ease text editing procedures, the CRTC allows the user to split each row into segments. This partitioning is necessary when dealing with window positioning within the screen. The "window" section provides detailed information. Each segment may contain up to 255 visible characters and up to 255 hidden characters limited by the eight-bit counter.

Hidden characters are characters that the CRTC fetches from system memory but that are not loaded into the internal row buffers. They are identified by the Ignore Bit of the attribute word when DH in Mode Register 1 is reset. An attribute flag character is also a hidden character if the Invisible Attribute Flag (IAF) of Mode Register 1 is set.

The background linked-list should be terminated with a Termination Row Control Block. This block consists of a Row Control Block Pointer pointing to itself, a Character Code Pointer set to "Ø",and C-flag=Ø.

RCB OVERVIEW:

RAØ,1 - A two-word link pointer pointing to the next Row Control Block.
RA2-6 - The first segment's block composed of 5 data words:
- The numbers of visible and hidden characters in the segment constitute the first data word,
- The segment's character-list pointer (nexttwo data words),
- The segment's attribute-list pointer (two words),
- Successive segments are identical to the first,
- An optional "Row Redefinition Block" pointer (two data words).

The user must set at least one Row Redefinition Block after power-up. A Row Redefinition Block contains characteristics applicable to a row. This information stays latched until another Row Redefinition Block is encountered. If no Row Redefinition Block is fetched after power up, information such as character start and end scan lines is undefined. If N segments are present in a Row Control Block, its length will be either:

N * 5 + 2 - if no Row Redefinition Block is present
N * 5 + 4 - if a Row Redefinition Block is present

RCB DETAILED DESCRIPTION:

RAØ,1 - The most significant bit in the first word indicates if a Row Redefinition Block has to be loaded for the current row. When this flag (LNK) is "1", then the Row Redefinition Block will be loaded. The

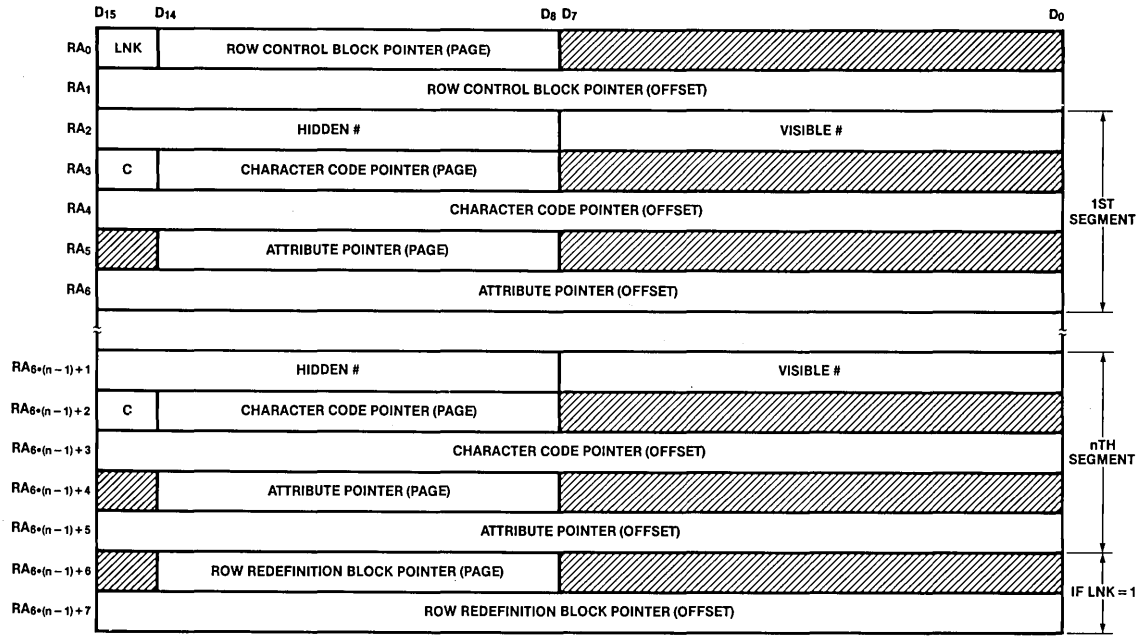


Figure 2.32. Row Control Block (L/$\overline{S}$ = 0)

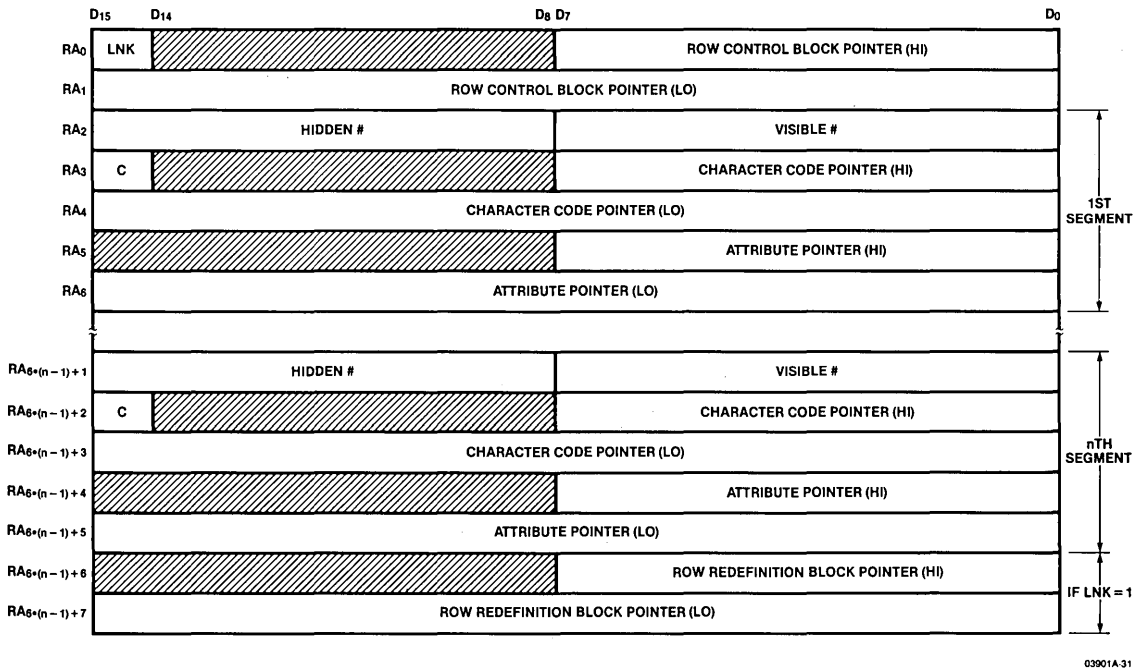


Figure 2.33. Row Control Block (L/$\overline{S}$ = 1)

remainder of the first two words contain the link pointer to next Row Control Block.

RA2 - The number of hidden characters and the number of visible characters are interpreted by the CRTC in the following way:

No window within the current row.
The DMA will use the sum of the hidden and visible character numbers to determine the number of characters to be fetched. In this case the CRTC does not distinguish between those two numbers; it uses only the sum.

Window within the current row.
If a window is present, both the number of hidden characters and the number of visible characters in background and window segments have to be specified correctly. The total number of hidden and visible characters determines the number of characters fetched from memory. The CRTC takes the number of visible charcters in the segment and the window coordinates of the Window Block to place the window.

RA3,4 - These two words contain the character code address pointing to the beginning of the character code string of this segment and the continue bit (C).
C = Ø This is the last segment of this row.
C = 1 The segment list continues.
If this pointer is "Ø", then the space specified by the visible number of characters for this segment is filled with the fill code.

RA5,6 - The pointer links to the attribute string of this segment.

The segment header (RA3-6) must be repeated for each additional segment. If the LNK-bit in RAØ is set, the two words following the last segment header must contain the pointer to the Row Redefinition Block

ROW REDEFINITION BLOCK:
The Row Redefinition Block is composed of five words. These words hold information relevant to the display characteristics of the row (Figure 2.34).

RRØ	:	Total Scan Line Count (TSLC)	5 Bits
		Normal Character Start (NCS)	5 Bits
		Normal Character End (NCE)	5 Bits
RR1	:	Row Attributes	5 Bits
		Superscript Character Start (SPCS)	5 Bits
		Superscript Character End (SPCE)	5 Bits

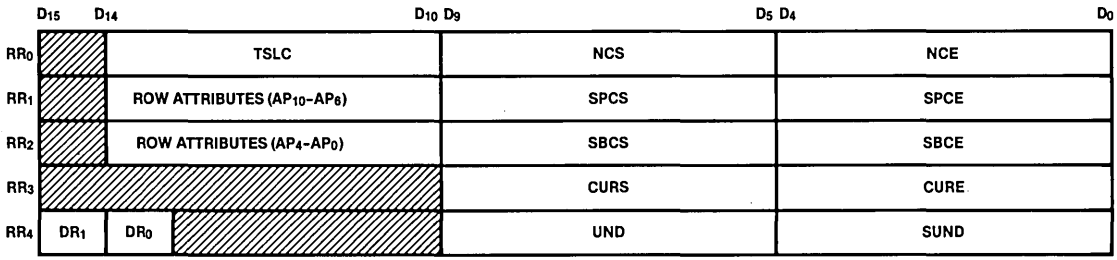


Figure 2.34. Row Redefinition Block

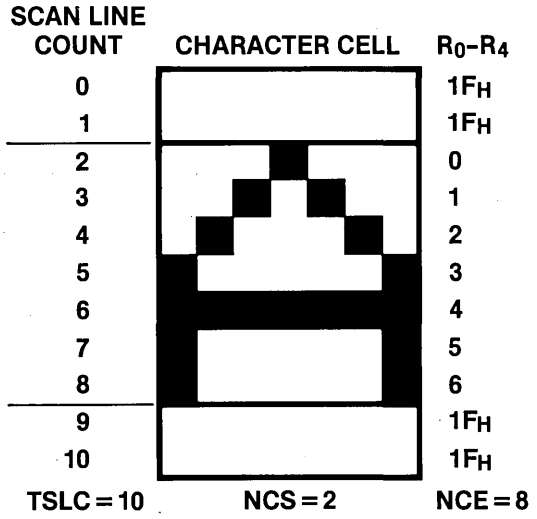


Figure 2.35. Character Placement

RR2	:	Row Attributes		5 Bits
		Subscript Character Start (SBCS)		5 Bits
		Subscript Character End (SBCE)		5 Bits
RR3	:	Cursor Start		5 Bits
		Cursor End		5 Bits
RR4	:	Double Row	(DR)	2 Bits
		Underline	(UND)	5 Bits
		Shifted Underline	(SUND)	5 Bits

All this information is captured by the CRTC. It acts on the invoking character row and succeeding ones until a new Row Redefinition Block is invoked.

The Total Scan Line Count (TSLC) defines the total number of scan lines per row minus one.

Normal Character Start (NCS) and End (NCE) define the vertical position and height of normal characters within the row.

Same definition applies to superscript and subscript characters with SPCS, SPCE, SBCS, SBCE.

When the scan line count is less than the character start scan line value (NCS, SPCS, or SBCS) or larger than the character end scan line value (NCE, SPCE, or SBCE), RØ-R4 puts out $1F_H$. Figure 2.35 shows an example. Normally the character slice with the address $1F_H$ is programmed to be blank.

More details concerning these parameters are included in the attributes section.

There are 10 user-definable row attribute bits which are output on the APØ-AP4 and AP6-AP10 pins during the horizontal retrace time. Bits D14 through D10 in RR1 are output on AP10-AP6, while bits D14 through D10 in RR2 are output on AP4-APØ. This row attribute can be registered externally to the CRTC with the falling edge of HSYNC. This feature can be used for a set of user definable attributes or to implement functions which are not directly supported by the CRTC; e.g., loadable character font generator or horizontal soft-scroll. Cursor start and cursor end applies to partial, reverse and underline cursors, and defines the position and height of the corresponding cursor. (See section "Cursor Display").

The Double Row bits (DR1, DRØ) allow the user to insert double row characters in the text on a row basis. The code is interpreted as follows:

DR1	DRØ	
0	0	Normal Character Row
0	1	Reserved
1	0	Top Half of a Double Row
1	1	Bottom Half of a Double Row

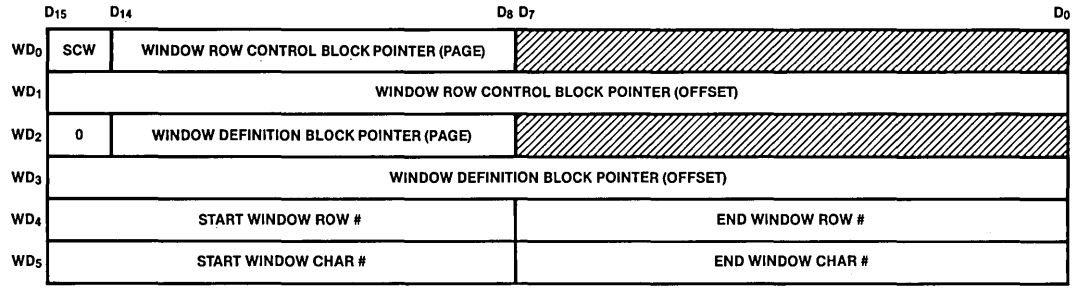


Figure 2.36. Window Definition Block (L/$\overline{S}$ = 0)

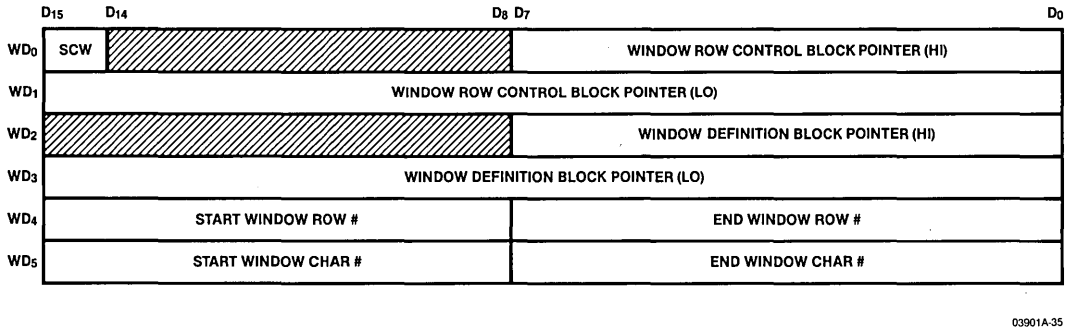


Figure 2.37. Window Definition Block (L/$\overline{S}$ = 1)

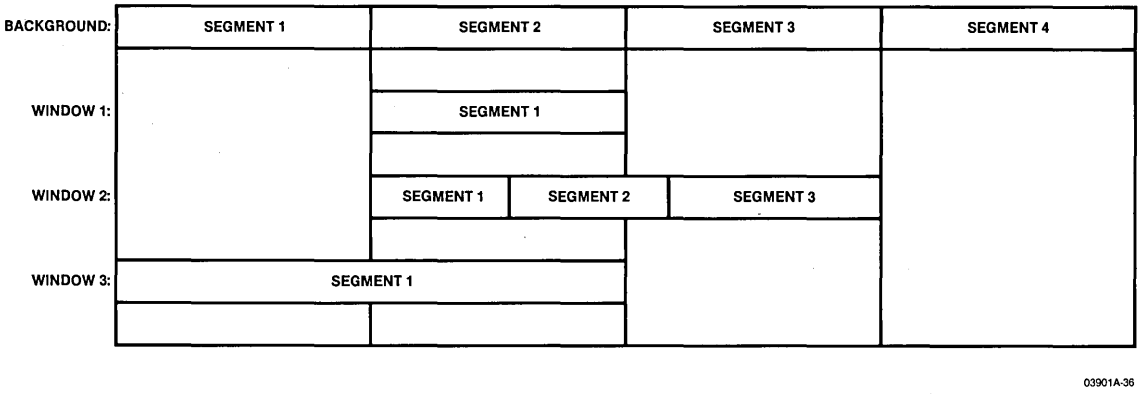


Figure 2.38. Window Overlay

The linked-list for a double size row consists of two Row Control Blocks, one for the top half of the row and one for the bottom half. The data accessed by these row control blocks should be identical, apart from the DR bits in the Row Redefinition Blocks.

Underline specifies the scan line number on which the underline attribute will act.

Shifted underline acts the same way as underline except that it applies to the shifted underline attribute.

2.5.4 Window Information Management

The Top Of Window register (TOW) points to the first word of a Window Definition Block (WDB), which specifies the window characteristics. There is one Window Definition Block per window, and they are linked together starting with the topmost window on the screen (whose WDB is pointed by TOW). If TOW=0, no window is displayed on the screen.

The Window Definition Block defines the following parameters (see Figures 2.36 and 2.37):

WDØ,1 - First Window Control Block link-pointer (2 words)
WD2,3 - Next Window Definition Block link-pointer (2 words)
WD4 - The start and end window row numbers (1 word)
WD5 - The start and end window character numbers (1 word)

The Window Row Control Block Pointer points to the Window Row Control Block specifying the first row of the window. The most significant bit of WDØ (Soft-Scroll Window, SCW) indicates if this particular window should be scrolled:

SCW	Soft Scroll Window
Ø	Window Soft Scroll Disabled
1	Window Soft Scroll Enabled

Note that the soft-scrolling does not occur until conditions specified in the Main Definition Block are satisfied.

When the pointer to the next Window Definition Block is equal to zero, there are no more windows on the screen. Otherwise, the pointer indicates the address of next Window Definition Block.

The start and end window row numbers are two bytes which indicate the vertical position of the first and last window rows on the screen expressed in row number.

The most significant bit of WD2 must be "Ø" when L/S = Ø.

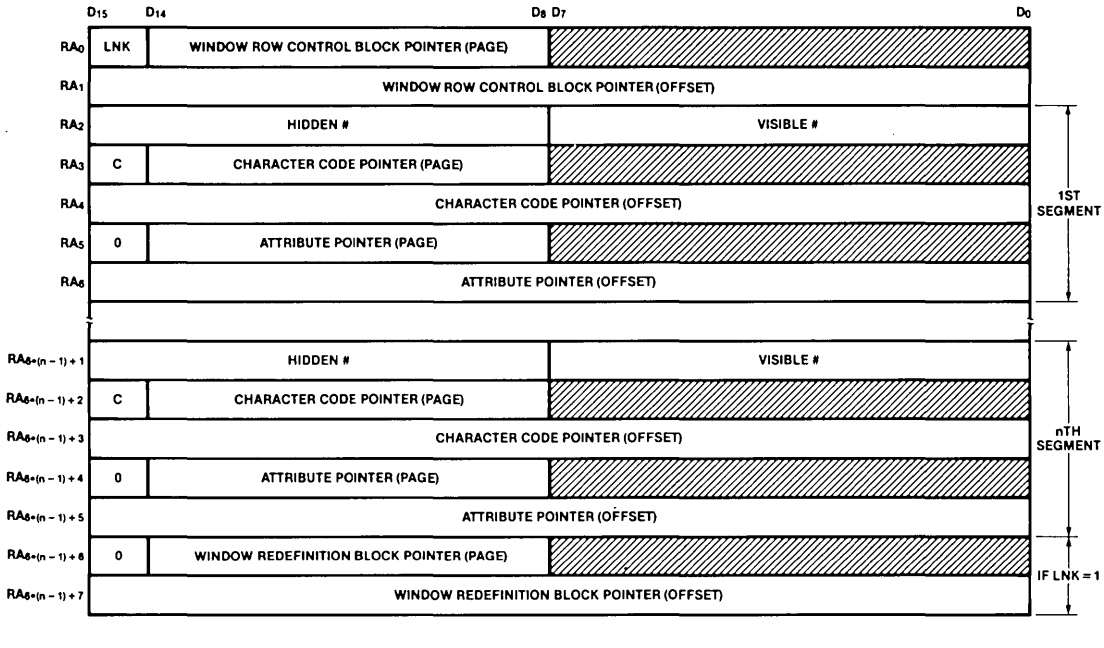


Figure 2.39. Window Row Control Block (L/$\overline{S}$ = 0)

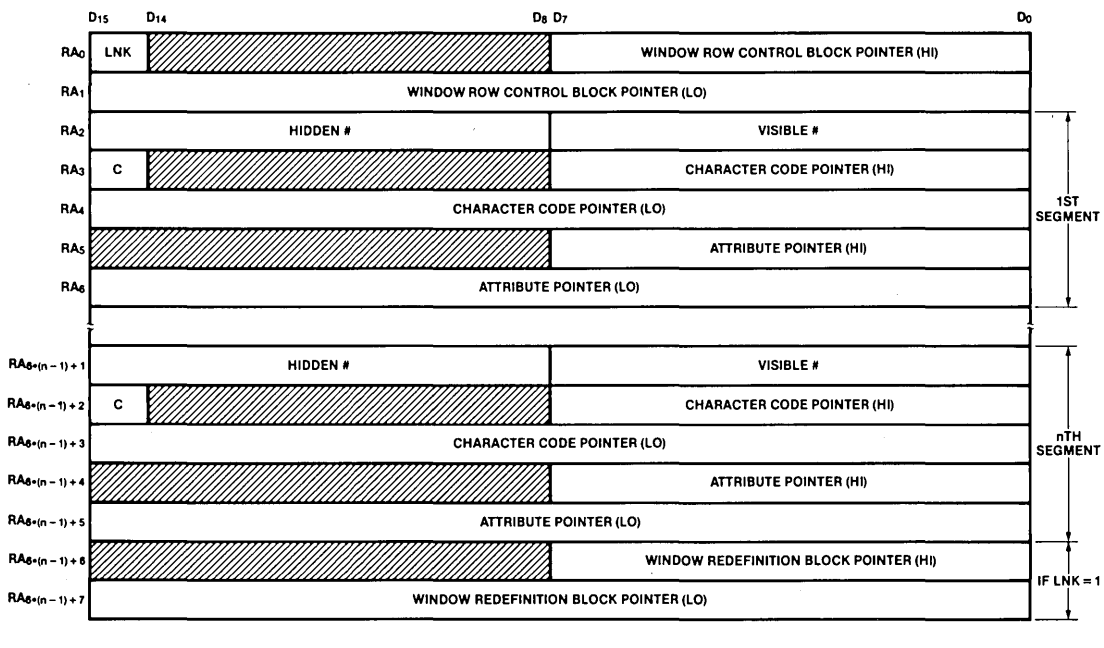


Figure 2.40. Window Row Control Block (L/$\overline{S}$ = 1)

The start and end window character numbers are two bytes which indicate the horizontal position of the first and last window characters on the screen.

As mentioned above, the Window Control Block is identical to the Row Control Block (Figure 2.39 and 2.40). However, some restrictions should be observed when dealing with windows:

- The number of visible characters of the overwritten background segment is effectively interpreted by the row management unit whenever a window is present within the row. When no window is present, the CRTC needs only the sum of hidden and visible characters of the loading segment to know the length of the segment in memory.

- The start and end positions of the window have to match segment boundaries in the background display. A window may span multiple segments (see Figure 2.38).

- Only one window can exist between the row numbers specified by start window row # and end window row #.

- When the contents of a window row's linked-list do not fill the window's row, the fill code will be used to fill the remaining character positions of that window's row. During that time, the bus is not released and dummy DMA cycles are executed.

- The Window Redefinition Block (Figure 2.41) is structured similarly to the Row Refinition Block. TSLC is left out, since a window row has to have the same number of scan lines as the background row.

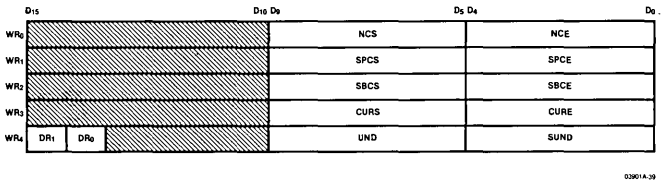


Figure 2.41. Window Redefinition Block

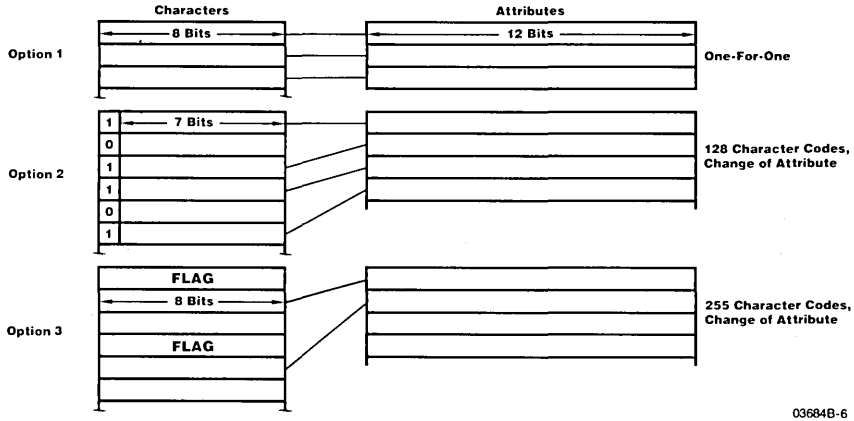


Figure 2.42. Attribute Fetch Options

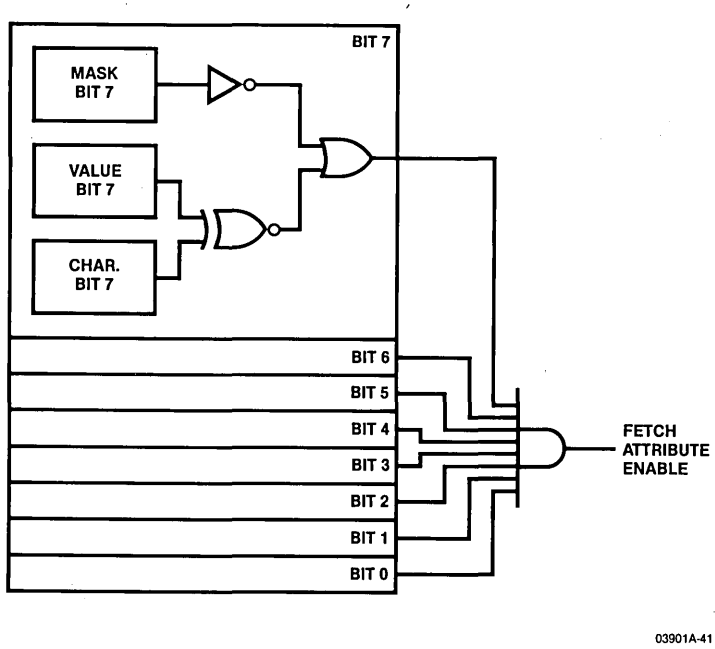


Figure 2.43. Attribute Flag Detect Mechanism

2.6 ATTRIBUTES

In this chapter we will focus on the character attribute architecture and the various display operations that can be handled by the CRTC. Since the user may have very specific display requirements that match with his own design, the CRTC has been designed to provide great versatility in the attribute options.

In the character stream two pieces of information are present:
- The actual character code
- An attribute invoking flag that may be part of the character code, or a specific code by itself. This option is programmed via the attribute-flag register internal to the CRTC. The function of this register is described in the Register Description section.

Once the choice of attribute-invoking flag(s) has been made, it is possible to either display or inhibit the display of the flag by using the invisible attribute flag (IAF) bit contained in Mode Register 1. If IAF=Ø, each code invoking an attribute is displayed, meaning that this specific code will not only invoke an attribute, but also be output on CCØ-CC7 to address the character generator. This character will be affected by the invoked attribute. If IAF=1, any code invoking an attribute is not loaded into the row-buffer and the invoked attribute will then affect the following character. If two or more successive flags are present in the stream, only the last one (and the attribute it invokes) will affect the first displayable character code encountered (see Figure 2.42). Figure 2.43 shows the Attribute Flag detect mechanism.

A character attribute is a code which will affect the display characteristics of a character or set of characters on the screen.

We can distinguish four levels of attributes:
- character attributes
- field attributes
- row attributes
- frame attributes

2.6.1 Character Attributes

Character attributes are word quantities which affect various CRTC output signals and other operations on a character by character basis.

These words reside in memory and are accessed via the attribute-segment pointers associated with the character-segment pointers in the Row Control Blocks.

The character attributes are stored in parallel with the corresponding character code in each row buffer. The bits in the attribute word are discussed below:

ATTRIBUTE DESCRIPTION

Bit 15	:	Latched/Unlatched	Bit 7:	User Definable
Bit 14	:	Cursor	Bit 6:	Highlight
Bit 13	:	Ignore Character	Bit 5:	Reverse
Bit 12	:	Reserved	Bit 4:	Superscript
Bit 11	:	Reserved	Bit 3:	Subscript
Bit 10	:	User Definable	Bit 2:	Shifted Underline/Strike Through
Bit 9	:	User Definable	Bit 1:	Underline
Bit 8	:	User Definable	Bit Ø:	Blink

The Attribute Port Enable and Attribute Redefinition Register affect the attribute processing. Refer to the "Register" Section.

BLINK
When this bit is set in the attribute word, the APØ pin will output a periodic signal whose rate and duty cycle are specified in the Main Definition Block. When this bit is reset, APØ outputs a Low level. Blink may be programmed to be a user-definable attribute. In this case, no internal blink attribute processing is done.

UNDERLINE
When this bit is set in the attribute word, the AP1 pin will output a High for one scan line in the character cell. The scan line on which the underline is active is specified in the Row Redefinition Block and can, therefore, be changed on a row-by-row basis. If this attribute is made user-definable (see Attribute Redefinition Register), the pin is active for all scan lines of the character cell. Underline is active for two scan lines when displaying double-height rows.

SHIFTED UNDERLINE
This bit acts like Underline except that the signal is output on AP2 and the scan line number is specified by an independent 5-bit word also contained in the Row Redefinition Block.

SUBSCRIPT
When this bit is set, the affected character will be displayed on a set of scan lines specified by subscript character start line # and subscript character end line # in the Row Redefinition Block. This bit is generally used to display subscript characters. In addition to this internal process, a High level is output on AP3 indicating a subscript character. This feature may be used to switch to a different character font generator. The subscript attribute pin is active for all scan lines between start line # and end line #. If it is programmed to be a user-definable attribute, the pin is active for all scan lines of the character cell.

SUPERSCRIPT
Similar to subscript. The set of scan lines is specified by superscript character start line # and superscript character end line # in the Row Redefinition Block. The attribute is output on AP4. It can also be programmed to be a user-definable attribute.

REVERSE
When this bit is set, a High level is output on AP5. This bit may be used to reverse the invoking character on the screen. No internal attribute processing is done, so this attribute can be treated as a user-definable one. Reverse is exclusive ORed with the reverse cursor.

HIGHLIGHT
When this bit is set, a High level is output on AP6. This bit may be used to highlight the invoking character on the screen. No internal attribute processing is done, so it can be treated as user definable if desired.

USER-DEFINABLE
These four bits have their state output on the matching pins (AP7-AP10) and can be used as desired to affect the invoking characters.

IGNORE CHARACTER
When the ignore bit is set to "1", and the display hidden (DH) bit in Mode Register 1 is reset (Ø), neither the affected character nor its attribute code are loaded into the row buffer and thus are not displayed. When DH is set, the ignore characters (those having invoked the ignore attribute) are loaded along with their attribute code. The ignore bit is not put out on the attribute port.

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

LATCHED/UNLATCHED
When this bit of the attribute word is set ("latched") the attribute information applies to all characters following the character that invoked the attribute word. This will be described in more detail in the field attribute paragraph. This bit is not put out on the attribute port.

CHARACTER ATTRIBUTE TIMING:

The attribute information present on the attribute port is output coincident to or one character clock after the invoking character, depending upon the skew-bits in Mode Register 1.

ATTRIBUTE PORT ENABLE REGISTER:

The function of this register is described in the Register Description Section. The superscript and subscript effect will not be cancelled by resetting the corresponding bits in this register; in fact, this will only drive the corresponding attribute port pins Low. The internal attribute processing still takes place. To disable subscript and superscript action, the Attribute Redefinition Register must be used.

The subscript and superscript, when enabled, may be used to choose between a standard character generator and a specific character generator for subscript and/or superscript. However, in most applications, one standard font generator can be used for all three.

ATTRIBUTE REDEFINITION REGISTER:

Four user-definable attributes are provided for optional external attribute processing. If this number is not sufficient, then the highlight and reverse attributes may be used as user-definable without any modification.

If this is still not enough, the user can disable the normal effect of other attributes and turn them into user-definable attributes. These attributes are:

- superscript
- subscript
- shifted underline
- underline
- blink

This yields 11 user-definable attributes. The function of the Attribute Redefinition Register is described in the Register Description Section.

2.6.2 Field Attributes

A field attribute affects a set of successive characters. This feature reduces memory consumption and software complexity compared to character attributes when dealing with character strings. Field attributes are similar to character attributes and are implemented by setting the latched attribute bit.

When a character does not invoke an attribute, it implicitly invokes the previously latched attribute. Therefore, every character appearing on the screen is associated with an attribute:

- The character invoked a latched or unlatched attribute. This attribute will affect that specific character (if it is a displayable character).

- The character does not invoke an attribute. The last latched attribute will affect this character.

As specified earlier, when an ignore attribute is invoked and Display Hidden is reset, the attribute word and the character are not loaded in the row buffers. However, if the invoked attribute is a latched attribute, then the ignore will be latched and succeeding characters will not be loaded unless they invoke an attribute with ignore reset. This ignore effect will be cancelled for all succeeding characters as soon as a latched attribute with ignore bit reset is invoked.

A latched attribute affects all subsequent characters not involving attributes, whether they are in windows or background, until a new latched attribute is encountered. As a result a latched attribute wraps around the screen, ripples through rows, background-window and window-background, etc.

2.6.3 Row Attributes:

The row attributes are 10 bits that are output on APØ-AP4 and AP6-AP10 at horizontal retrace time.

This is a feature of the CRTC that enables the user to modify the display characteristics on a row-by-row basis.

The row attributes are specified in the Row Redefinition Block and may be latched by external logic at HSYNC fall time. Some examples of applications of row attributes follow. The shape of the modified area(s) will always be a horizontal screen slice(s):

- reverse row(s)
- highlight row(s)
- blink row(s)
- color palette addressing
- row(s) underline
- change character set
- switch to semi-graphic generator
- switch video output to a graphic
- display unit to mix graphic and text
- blank row(s) (secret prompts)

The Row Attributes are internally latched and do not need to be rewritten on each row. Therefore, the internal row attribute register is updated each time a Row Redefinition Block is invoked.

2.6.4 Frame Attributes

Frame attributes affect the character display characteristics of the entire screen. These attributes are stored in the Main Definition Block and define:

- x-y cursor positioning
- fill character code
- x-y cursor blink rate and duty cycle
- attribute cursor blink rate and duty cycle
- soft scroll of window or background
- soft scroll rate and direction

2.6.5 Cursor Display

Cursors are used to locate specific points in the text that need particular attention.

Two types of cursors are supported by the CRTC:
- an absolute cursor (x-y cursor)
- an attribute cursor

THE ABSOLUTE CURSOR:
This cursor is positioned on the screen accordingly to its X (horizontal) and Y (vertical) coordinates specified in the Main Definition Block, and fetched by the CRTC during the vertical retrace time.

X is expressed in character units. X=Ø indicates the first character column. Y is expressed in row units. Y=Ø indicates the first row on the screen. This cursor is called absolute because it refers to the screen boundaries and is not dependent upon the text displayed on the screen. When the text is scrolled, the cursor position stays stationary relative to the screen.

When the CRT monitor beam matches the cursor position, a CRTC internal cursor signal goes High to indicate a match. This signal may be steered internally to one of three output pins: cursor pin, reverse pin, and underline pin.

The choice of the output pin is made through the cursor mask contained in Mode Register 2. In the same register, a cursor enable bit, when reset, controls the disable the Absolute Cursor. Furthermore, it is possible to partially affect the character position on the screen by specifiying the scan line boundaries in which the output signal will be active. These boundaries are specified in the Row Redefinition Blocks by CURS and CURE.

THE ATTRIBUTE CURSOR:
This cursor is positioned with the visible character that invoked an attribute with Cursor Bit=1. A display screen can therefore contain as many attribute cursors as there are character positions.

An attribute cursor is implicitly linked to the text in which it is contained. If the text scrolls up, the attribute cursor will scroll with the text, whereas the absolute cursor would remain steady.

When an attribute cursor is encountered, the same operation as with the absolute cursor will occur. However, a different set of bits in the cursor mask register steers the attribute cursor signal to one of the three outputs. This allows the user to distinguish the attribute cursor from the absolute cursor on the screen. The same scan line boundaries are used for both cursors.

CURSOR CHARACTERISTICS:
One out of four shapes may be chosen for each of the two cursors described earlier:

-Cursor Whole
The cursor signal is output on the cursor pin for each scan line of the character position.

-Cursor Part
The cursor signal is output on the cursor pin for the specific scan lines contained between cursor start and cursor end boundaries specified in the Row Redefinition Block.

-Reverse
Same operation as cursor part except that the signal is output on the reverse attribute pin after being exclusive ORed with the internal reverse attribute signal.

-Underline
Same operation as cursor part except that the signal is output on the underline attribute pin.

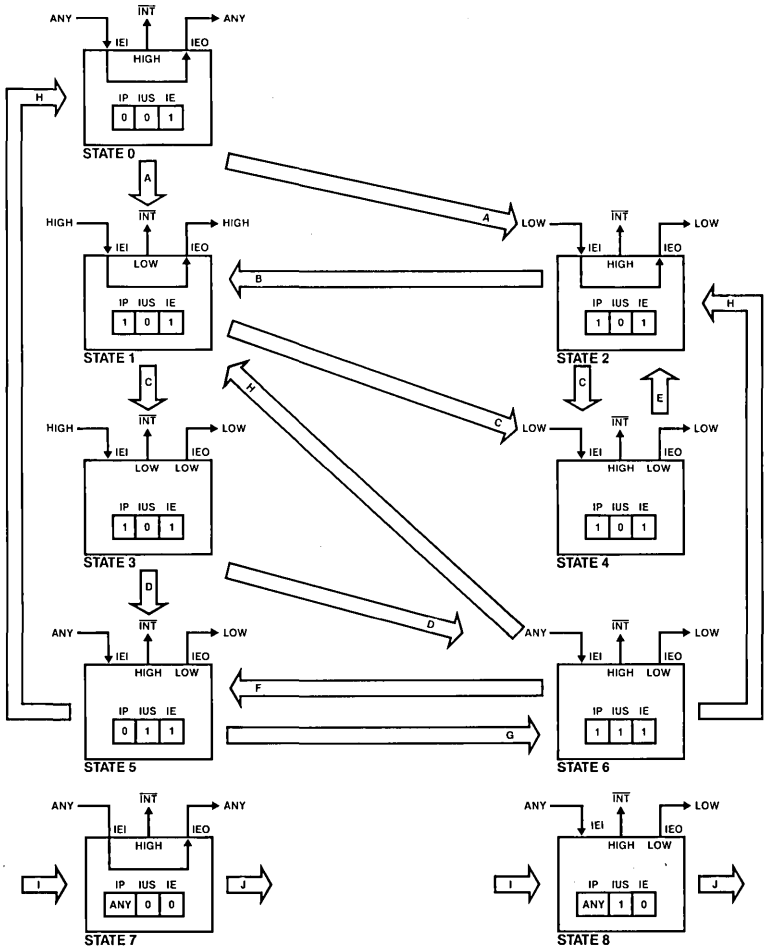
2.6.6 The Fill-Code Attributes

When the Row Management Unit reaches the end of the last segment of a row, and the row-buffer is not full (96 characters or 132 depending upon "slim" setting), the Row Management Unit will fill the remaining space in the row buffer with a specific code specified by the user in the Main Definition Block. This code is the fill code, and needs special attention when used in a text:

- Each time the row buffer is not filled by the contents of the linked-list, the fill code is loaded into the row buffer.

- If the fill code is an attribute invoking code, the Row Management unit may not invoke an attribute, depending on the "FAT" bit in the Main Definition Block. If the user needs to display the fill code associated with an attribute, he should then set the "FAT" flag (Fill Code Attribute in the Main Definition Block) to one and add the desired attribute in the attribute list of the last segment invoked. Only one attribute word will be fetched for the fill characters, so this attribute must be a latched attribute to affect all fill characters loaded into the row buffer.

- The ignore attribute will be discarded when associated with the fill code.



Transition Legend

- A: The peripheral detects an interrupt condition and sets Interrupt Pending.
- B: All higher priority peripherals finish interrupt service, thus allowing IEI to go High.
- C: An interrupt-acknowledge transaction starts, and the IEI/IEO daisy chain settles.
- D: The interrupt-acknowledge transaction terminates with the peripheral selected. Interrupt Under Service (IUS) is set to 1, and Interrupt Pending (IP) may or may not be reset.
- E: The interrupt-acknowledge transaction terminates with a higher priority device having been selected.
- F: The Interrupt Pending bit in the peripheral is reset by an I/O operation.
- G: A new interrupt condition is detected by the peripheral, causing IP to be set again.
- H: Interrupt service is terminated for the peripheral by resetting IUS.
- I: IE is reset to zero, causing interrupts to be disabled.
- J: IE is set to one, re-enabling interrupts.

State Legend

- 0: No interrupts are pending or under service for this peripheral.
- 1: An interrupt is pending, and an interrupt request has been made by pulling $\overline{INT}$ Low.
- 2: An interrupt is pending, but no interrupt request has been made because a higher priority peripheral has an interrupt under service, and this has forced IEI Low.
- 3: An interrupt-acknowledge sequence is in progress, and no higher priority peripheral has a pending interrupt.
- 4: An interrupt-acknowledge sequence is in progress, but a higher priority peripheral has a pending interrupt, forcing IEI Low.
- 5: The peripheral has an interrupt under service. Service may be temporarily suspended (indicated by IEI going Low) if a higher priority device generates an interrupt.
- 6: This is the same as State 5 except that an interrupt is also pending in the peripheral.
- 7: Interrupts are disabled from this source because IE = 0.
- 8: Interrupts are disabled from this source and lower priority sources because IE = 0 and IUS = 1.

1. This diagram assumes MIE = 1. The effect of MIE = 0 is the same as that of setting IE = 0.
2. The DLC bit does not affect the states of individual interrupt sources. Its only effect is on the IEO output of a whole peripheral.
3. Transition I to state 6 or 7 can occur from any state except 3 or 4 (which only occur during interrupt acknowledge).
4. Transition J from state 6 or 7 can be to any state except 3 or 4, depending on the value of IEI, IP, and IUS.

Figure 2.44. State Diagram for an Interrupt Source

2.7 INTERRUPT OPERATION

An interrupt may occur whenever the CPU needs to be notified of various events internal to the CRTC or that an operation has just been completed. There are two sources of CRTC interrupts:

- Vertical Interrupt
- Soft-scroll Interrupt

VERTICAL INTERRUPT
The vertical interrupt, if enabled, can be used as a real time interrupt by the CPU or it can be used as an indication that certain CRT updates should take place. The vertical interrupt is issued when the "n-th" character row has been loaded by the CRTC into its internal row buffers. The value of "n" is determined by the 8-bit VERTINT field in the HSYNC register. When "n" is set to "1", the CRTC issues a vertical interrupt after the last segment of the first row is completely loaded.

SOFT-SCROLL INTERRUPT
The soft-scroll interrupt is used to inform the CPU when to update the display linked-lists during soft-scrolling. See the section on Soft-scroll for more details.

INTERRUPT PROTOCOL
A complete interrupt cycle consists of an interrupt request by the CRTC followed by an interrupt acknowledge of the CPU (Figure 2.44). The request, which consists of $\overline{INT}$ being pulled Low by the CRTC, notifies the CPU that an interrupt is pending. The interrupt acknowledge cycle notifies the peripheral that its interrupt has been recognized. In return, the peripheral may provide an interrupt vector to the CPU to identify itself (see the section on Row Management Unit.).

The CRTC has two sources of interrupt and each interrupt source has 3 bits that control the issuance of an interrupt. These bits are the Interrupt Pending bit (IP), the Interrupt Enable bit (IE), and the Interrupt Under Service bit (IUS). In addition to the control bits, two further bits control the interrupt behavior of the CRTC. These are the Disable Lower Chain bit (DLC) and the No Vector bit (NV) in Mode Register 2.

Peripherals are connected together via an interrupt daisy-chain formed with their IEI (Interrupt Enable In) and IEO (Interrupt Enable Out) pins. The daisy-chain resolves the interrupt priority.

For the purpose of this description, the CRTC may be considered as having two interrupt sources: Smooth Scroll, and Vertical Interrupt. The Smooth Scroll Interrupt has higher priority.

Figure 2.44 is a state diagram of interrupt processing for an interrupt source (assuming its IE bit is 1). An interrupt source with an interrupt pending (IP=1) makes an interrupt request (by pulling $\overline{INT}$ Low) only if it does not have an interupt under service (IUS=Low), no higher priority interrupt is being serviced (IEI=High), and no Interrupt Acknowledge transaction is in progress. IEO is not pulled down by the interrupt source at this time. IEO continues to follow IEI until an Interrupt Acknowledge occurs. Some time after $\overline{INT}$ has been pulled Low, the CPU initiates an Interrupt Acknowledge bus cycle. Between the falling edge of $\overline{INTACK}$ and the falling edge of $\overline{DS}$, the IEI/IEO daisy-chain settles. $\overline{AS}$ is optional. Any interrupt source with an interrupt pending (IP=1) holds its IEO line Low during Interrupt Acknowledge. All other interrupt sources make IEO follow IEI. When $\overline{DS}$ falls, only the highest priority interrupt source with a pending interrupt (IP=1) has its IEI input High and its IUS bit set at "Ø". This is the interrupt source being acknowledged and, at this point it sets its IUS bit to 1. If the peripheral's NV bit is "Ø", the interrupt source identifies itself by placing the interrupt vector on ADØ-AD7. Each time $\overline{DS}$ is activated during Interrupt Acknowledge cycles, the vector is put out. The upper byte is driven Low. If the NV bit is "1", the peripheral's AØ-AD15 pins remain floating, thus allowing external circuitry to supply the vector.

While an interrupt source has an interrupt under service (IUS=1), it prevents all lower priority devices from requesting interrupts by forcing IEO Low. When interrupt servicing is complete, the CPU must reset the IUS and the IP bits.

A peripheral's Interrupt Enable bit (IE) modifies the peripheral's behavior in the following manner - if the IE bit is "Ø", the effect is as if all interrupts from the peripheral are disabled. However, the peripheral can still set its IP bit if an interrupt is required. If the IE bit is cleared while the source is driving $\overline{INT}$ Low, $\overline{INT}$ will return High until IE is set. To prevent race conditions, the CPU should mask out interrupts from the peripheral before clearing IE. Note that IE, when cleared, also prevents the CRTC from responding to an Interrupt Acknowledge. While IE is cleared, IEO will follow IEI. The peripheral's IEO line can be forced unconditionally into the Low state by setting the DLC bit to "1".

2.8 SOFT-SCROLL MECHANISM

The Am8052 provides very powerful soft-scroll capability with minimum interaction by the CPU.

Window(s) or background can be soft-scrolled either up or down at a rate that is programmable via the scroll parameters field in the Main Definition Block.

Since the CRTC is architected to work with a linked list structure, some precautions should be taken when relinking the text after each scrolled row.

GENERAL SOFT-SCROLLING RULES:

Either windows or background can be scrolled at one time; they cannot be scrolled at the same time.

When a window splitting the screen vertically (sharing the row buffer with background characters) is intended to be soft-scrolled, then all of its rows must have the same total scan line counts (TSLC).

DOUBLE BUFFERING TECHNIQUE:

Soft scrolling operation is achieved by moving the appropriate data up or down on a scan line basis. Therefore, the CRTC adds an offset to the internal row's scan line count and outputs the result on RØ-R4. This results in a displacement of the data on the screen by an amount of scan lines equal to the offset. As soon as the end scan line (top or bottom depending on the scroll direction) of the first text row has reached the top extremity of the screen, a text relink has to be made. This relink serves to push the disappearing row off the screen or to link a new row onto the top of the screen.

In order to maintain a smooth relink transaction and allow for CPU time constraints, the Am8052 controls the relink timing through interrupts and double buffering of pointer register. As soon as the CRTC has begun soft-scrolling a character row, it generates an interrupt. The CPU which maintains the linked-lists responds by writing to "Top of Page (Window) Soft" a pointer value that will provide the correct linked-list for the display after it has completed the scroll of the current row. The CRTC will use this new value as the active "Top of Page (Window)" only after the row scroll in progress is completed. This double buffering of the "Top of Page (Window)" values allows maximum time (one character row scroll time) for the CPU to relink and respond to the interrupt.

According to the preceeding, when the user wants to soft-scroll a portion of the display (background or window), he should define two Main/Window Definition Blocks, and flip between those two blocks each time a soft-scroll interrupt occurs. This technique allows the user to execute the link modifications on the unused definition block while the other is being processed by the CRTC.

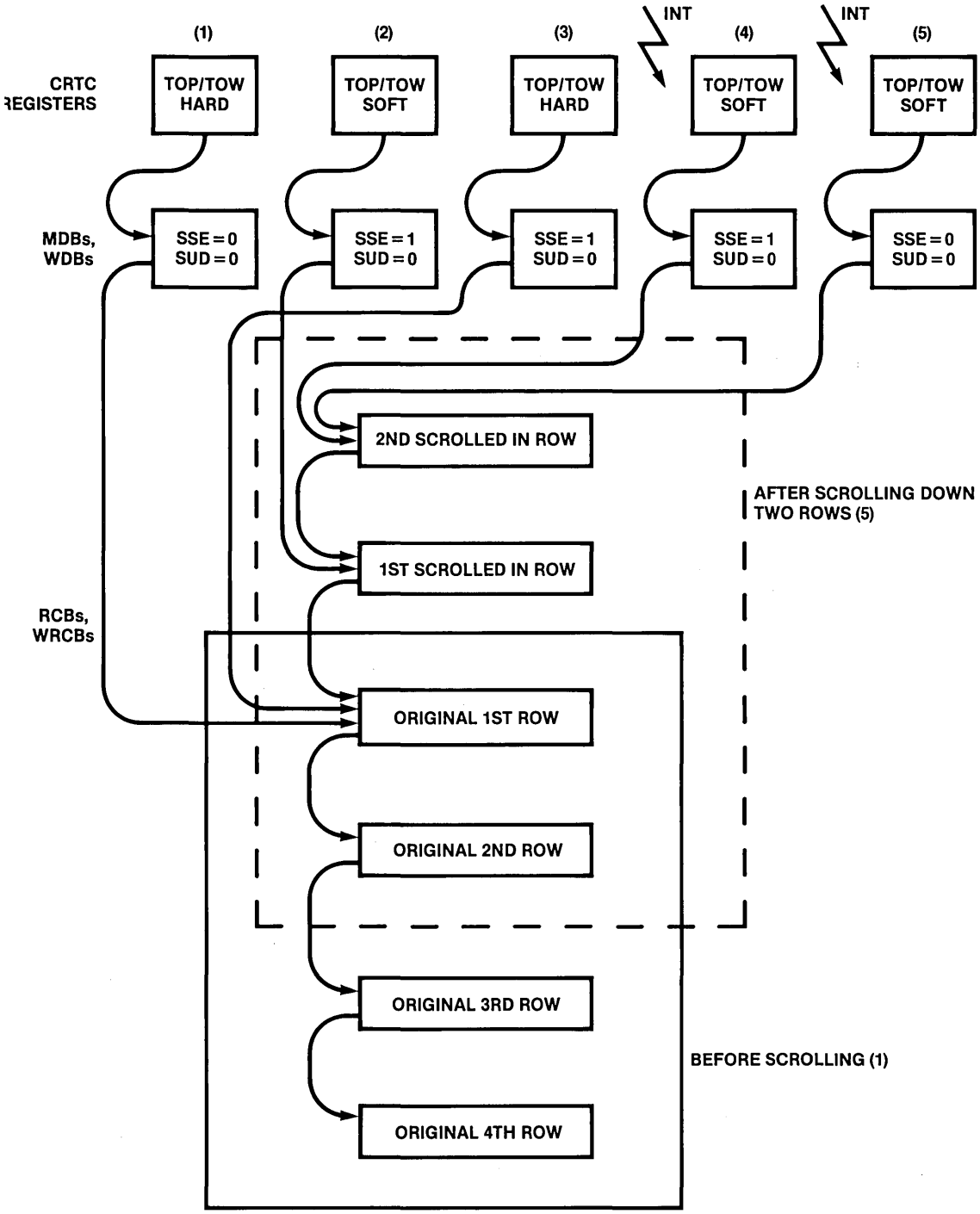


Figure 2.45. Scroll Down Sequence

03901A-43

DETAILED INTERLOCK MECHANISM:

The Top of Page/Window Soft is the key interface between the CPU and the CRTC when dealing with soft-scrolling.

When the CPU writes a pointer value into this register, it will not modify the actual Top of Page/Window register used by the CRTC to fetch the Main/Window Definition Block. In fact, the transfer beteen this temporary register to the actual register will take place according to the soft-scroll algorithm internal to the CRTC. Therefore, if the soft-scroll process has not been enabled, writing to Top of Page/Window Soft will not change anything to the link architecture and this register should be used only if soft-scroll operation is (or will be) performed. If the user wants to change the link in a non-soft-scroll condition he should use the "Top of Page/Window Hard" register.

The soft-scroll mechanism is enabled by setting the Soft-Scroll Enable bit (SSE) in the Main Definition Block. Two other bits in the Main Definition Block are used to select Window/Background scrolling and Up/Down scrolling directions. Additionally, when scrolling windows, the Soft-Scroll Window bit (SCW) in the corresponding Window Definition Blocks must be set. All windows which have SCW set are scrolled simultaneously. Windows which have SCW reset remain steady.

SCROLL DOWN:

The "Top of Page/Window Hard" register links to the Main/Window Definition Block of the currently displayed text. When a down scroll is initiated, the current text is moved down a fraction of a row. The empty space at the top of the screen is filled with a fraction of the scrolled-in row. Therefore, the CRTC has to know the pointer to the new Main/Window Definition Block, before it can start scrolling. The pointer is loaded into the "Top a Page/Window Soft" register.

DOWN SCROLL SEQUENCE:

The programming sequence shown in Figure 2.45 refers to both scrolling background or windows. The example shows two rows scrolling in a background or window consisting of a total of 4 rows. When scrolling the background the TOP Soft register is reloaded and two Main Definition Blocks are used to implement the "Double Buffer" technique. If a window is scrolled, the TOW Soft register and two Window Definition Blocks are involved. The numbers in this programming sequence below corresponds to Figure 2.45.

1. The CRT system displays a steady screen. The TOP/TOW Hard register links to a MDB/WDB with soft-scroll disabled. The soft-scroll process is initiated from this steady state.

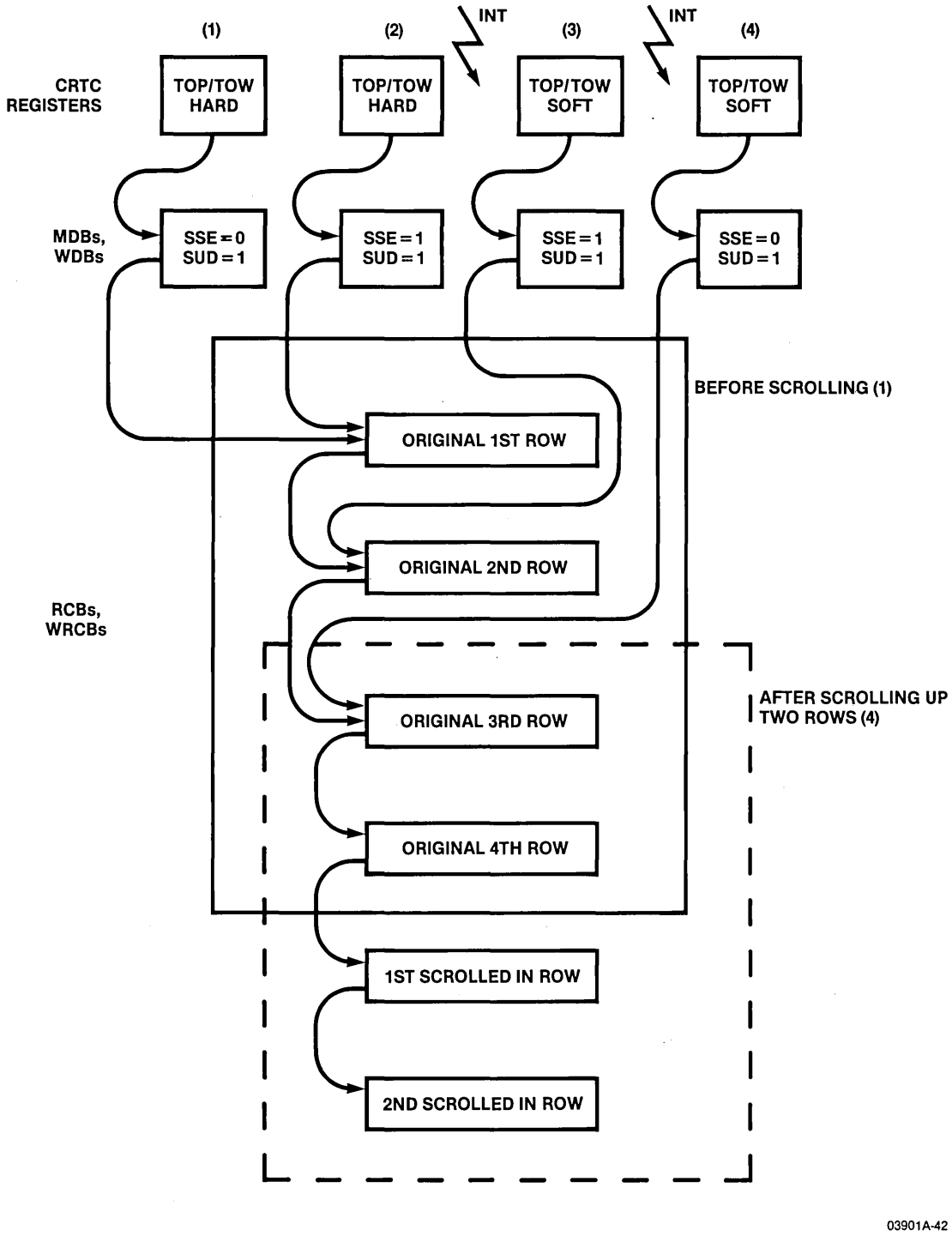


Figure 2.46. Scroll Up Sequence

2. The CPU prepares another MDB/WDB with soft-scroll enabled. This MDB/WDB contains a pointer to the RDB/WRCB for the scrolled-in row which in turn points onto the first row currently displayed on the screen. The CPU loads the pointer to this MDB/WDB into TOP/TOW Soft register.

3. The CPU then enables soft-scrolling by setting the soft-scroll bit in the MDB/WDB involved in Step 1. The CRTC detects this change when it fetches this block during the next vertical retrace period. The first frame after this change will still reflect the same linked-list, and the scrolling begins with the following frame. If the TOP/TOW Soft register was not initialized, the start of scrolling waits for the initialization. At this time, the CRTC transfers the contents of the TOP/TOW Soft register to the TOP/TOW Hard register to allow scrolling in the new row. It issues an interrupt on soft-scroll event to notify the CPU that the TOP/TOW Soft register can be updated. The update can take place at any time until the new row is entirely scrolled-in. If the update was not performed at that time, the displayed text will scroll up (hard scroll) one row and this same row will be soft-scrolled in again.

4. The TOP/TOW Soft register is relinked to the MDB/WDB pointing to the RDB/WRCB of the next row to be scrolled-in. If only one row should be scrolled, Step 4 is left out. For scrolling "n" rows, Step 4 is repeated after each interrupt issued by the CRTC "n-1" times.

5. To stop the soft-scroll process, the new pointer in the TOP/TOW Soft register points to a copy of the previous MDB/WDB in which the SSE-bit is cleared.

UP SCROLL SEQUENCE:

The numbers in the programming sequence below correspond to Figure 2.46.

1. The TOP/TOW Hard register links to the MDB/WDB of the currently displayed text. Soft-scroll is disabled.

2. The scroll process is initiated by enabling soft-scrolling in this MDB/WDB. The TOP/TOW Soft register does not need to be loaded at that time. The last row displayed links to the row to be scrolled-in. The CRTC detects the change of the scroll enable bit when it fetches the block during the next vertical retrace period. After it has started soft-scrolling, it issues an interrupt on soft-scroll event to make the CPU update the TOP/TOW Soft register.

3. The TOP/TOW Soft register links to the MDB/WDB pointing to the RCB/WRCB of the row following the scrolled-out row. If only one row should be scrolled, Step 3 is left out. For scrolling "n" rows, Step 3 is repeated "n-1" times.

4. To stop the soft-scroll process, the TOP/TOW Soft register points to a MDB/WDB with scroll disabled.

The "SIP" Bit:

The "Scroll in Progress" bit (SIP) is a status bit in the Mode Register 2 indicating to the CPU that the CRTC is actually scrolling either window or background while SSE bit (Soft-Scroll Enable) is set. The SIP bit is set as soon as the CRTC has loaded the Main Definition Block with SSE=1. Nevertheless, once the CPU resets SSE to "0",the CRTC will wait until the entire soft-scroll is finished before resetting SIP to "0". Furthermore, when using vectored interrupt, the SIP bit appears in bit 1 of the interrupt vector and, therefore, allows the user to access two different programs with regard to soft-scroll status without polling the SIP bit.

Soft-scroll Parameters Location and Functions:

IUSS:
Interrupt Under Service for Soft-Scroll operation (Bit 2 in Mode Register 2) set by the CRTC.

IES:
Interrupt Enable Soft-Scroll bit 1 in Mode Register 2. Set by the CPU.

IPS:
Interrupt Pending for Soft-Scroll event. Bit Ø in Mode Register 2. Set by the CRTC. Reset by the CPU.

SIP:
Scroll in Progress, bit 8 in Mode Register 2. Set and reset by the CRTC.

2.9 SYNCHRONIZATION

The CRTC has two built-in synchronization mechanisms: External SYNC (ESYNC) and Reset for Test ($\overline{RSTT}$). These mechanisms are activated by applying signals to the synchronization input pins (ESYNC and $\overline{RSTT}$). The ESYNC input synchronizes the CRTC to an external frame frequency. In most applications this input locks the vertical timing to the power-line frequency to avoid screen swimming. $\overline{RSTT}$ synchronizes multiple CRT controllers.

MULTIPLE CRT CONTROLLER SYNCHRONIZATION:
The Reset for Test ($\overline{RSTT}$) input synchronizes two or more CRTCs. This synchronization sequence is executed only upon system initialization. Figure 2.47 shows the timing diagram.

The sequence of operation for $\overline{RSTT}$ is:

- Reset all CRTC's by pulling Reset ($\overline{RST}$) Low for at least five clock cycles (CLK1 or CLK2, whichever is slower).

- After $\overline{RST}$ becomes inactive, initialize all CRTC registers including Mode Register 1 and 2 with DE=Ø.

- Activate $\overline{RSTT}$ synchronous to CLK1 or CLK2 depending on the CLK1/2-bit in Mode Register 1. It must be synchronous to the clock determining the frame timing. It must meet the setup time t_s to avoid metastable problems.

- Reload Mode Register 1 and 2. Set DE=1 (Mode Register 1).

- Deactivate $\overline{RSTT}$ synchronous to CLK1 or CLK2. $\overline{RSTT}$ must be active for a minimum of five clock cycles and its rising edge must meet the hold time (t_h) requirement. The rising edge of $\overline{RSTT}$ triggers all CRTC's to start display synchronously.

EXTERNAL SYNC OPERATION:
The ESYNC input allows synchronization of the CRT display vertical frame rate to the power line frequency to eliminate waviness and other effects. The ES bit in Mode Register 1 defies whether ESYNC controls the Vertical Sync rate.

ESYNC is recognized by the CRTC for every field or frame. It causes the VSYNC signal to become active at the occurrence of HSYNC. In non-interlaced mode, VSYNC becomes active at the first rising edge of HSYNC following ESYNC's rising edge (Figure 2.48). In interlaced mode, VSYNC comes active at the next HSYNC active when in the even frame, or in the middle between two HSYNC's in the odd frame (Figure 2.49).

The VSYNC and HSYNC are inactive (BLANK is active) before, during, and after reset. When the display is enabled via mode bit DE, HSYNC output becomes active, while VSYNC waits for ESYNC active. The display is delayed up to one ESYNC period.

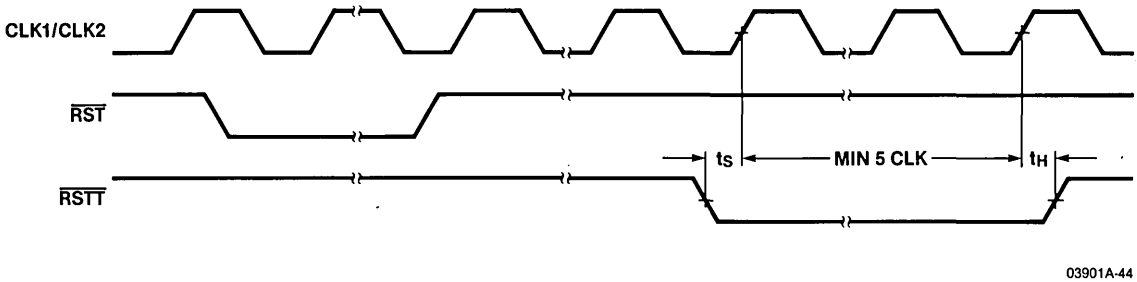


Figure 2.47. Reset for Test Timing

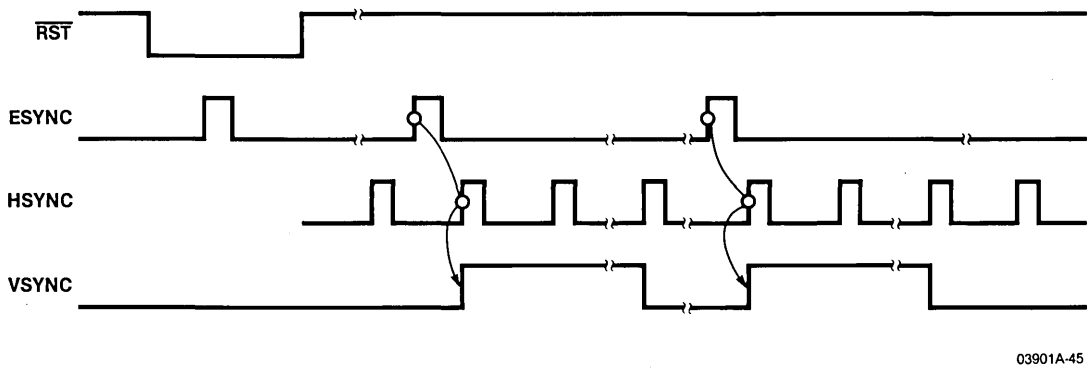


Figure 2.48. Non-Interlaced ESYNC Operation

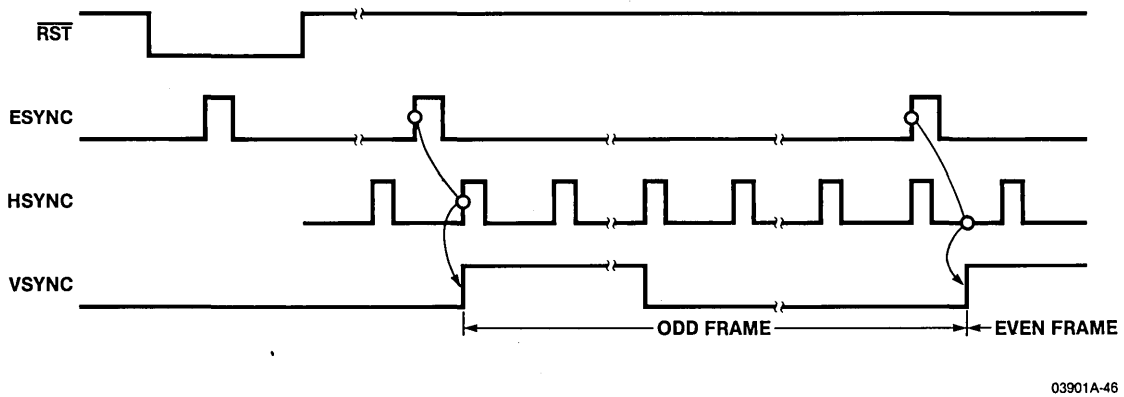


Figure 2.49. Interlaced ESYNC Operation

2.10 RFI AND INTERLACE VIDEO

There are two types of interlace, Repeat Field Interlace (RFI) and Interlaced Video (IV). Both types use the same vertical and horizontal timing as described in the Vertical and Horizontal Timing Section. 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 eliminate the horizontal stripes that sometimes occur between scan lines of non-interlaced displays (see Figure 2.50).

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-interlaced video. One problem with IV is the potential imbalance of CRT beam current between the odd and even fields and the resulting loss of perfect video interleave. This imbalance is greatest if the character rows consist of an even number of scan lines (odd field + even field). The CRTC averages the beam current for both fields if an odd number of scan lines is specified in TSLC for each character row and will alternately address the character ROM with all the even, then all the odd addresses on a Row by Row basis.

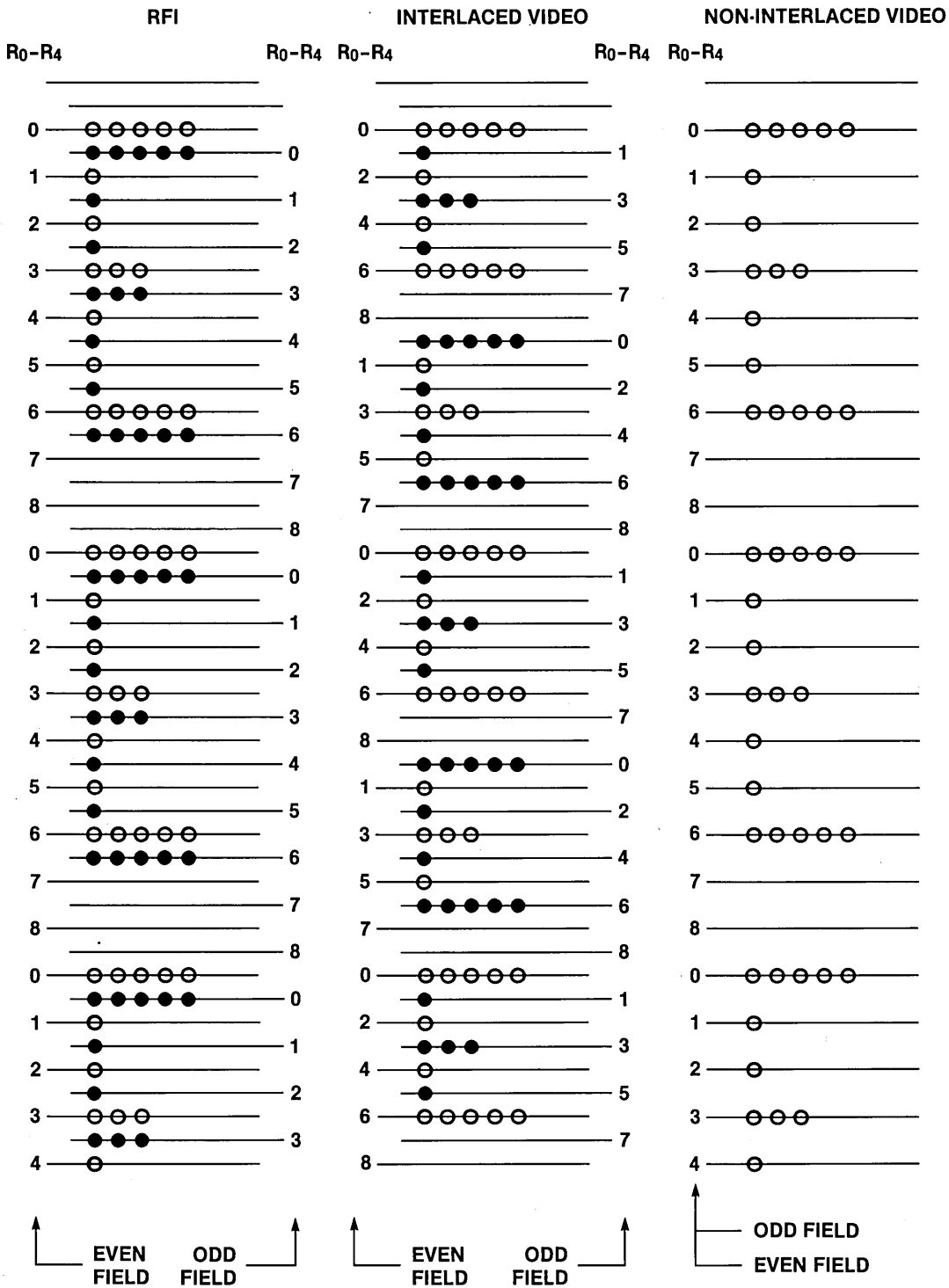


Figure 2.50. Scan Line Addressing

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 = 0 to +70°C; V_{CC} = 5V ±10%
C_{LOAD} = 15pF (All inputs except CLK_1 and CLK_2)
C_{LOAD} = 80pF (CLK_1 and CLK_2)

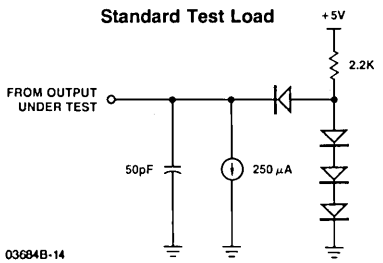
Parameters	Description	Min	Max	Units
V_{OH}	Output High Voltage	2.4		V
V_{OL}	Output Low Voltage (I_{OL} = 3.2mA)	0	0.4	V
V_{IH}	Input High Voltage (except CLK_1 and CLK_2)	2.0	V_{CC} +0.5	V
V_{CIH}	CLK_1/CLK_2 Input High Voltage	4.0		V
V_{IL}	Input Low Voltage (except CLK_1 and CLK_2)	−0.5	0.8	V
V_{CIL}	CLK_1/CLK_2 Input Low Voltage		0.3	V
I_{IX}	Input Load Current		±10	μA
IO_2, I_0	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

ORDERING INFORMATION

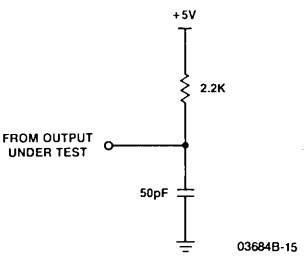
Package	Ambient Temperature	Clock 1 Frequency 6MHz	Clock 1 Frequency 8MHz
Leadless Chip Carrier	0°C ≤ T_A ≤ 70°C	Am8052-6LC	Am8052LC

Standard Test Conditions

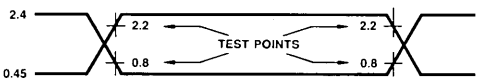
Standard Test Load



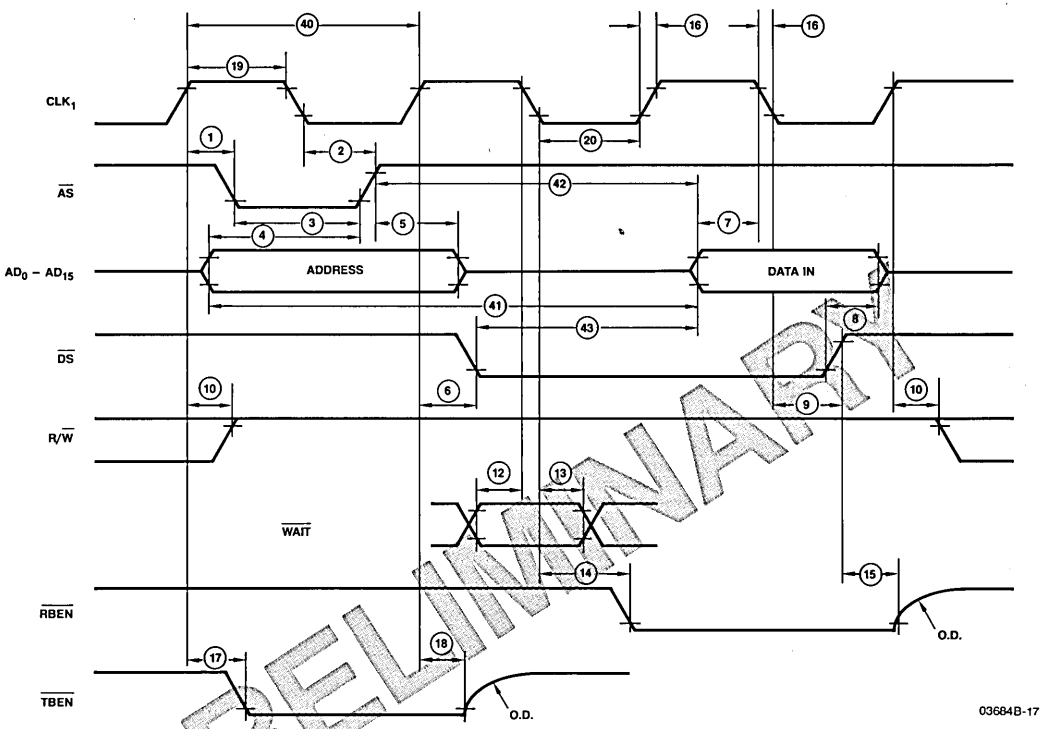
Open Drain Test Load



Input Waveform

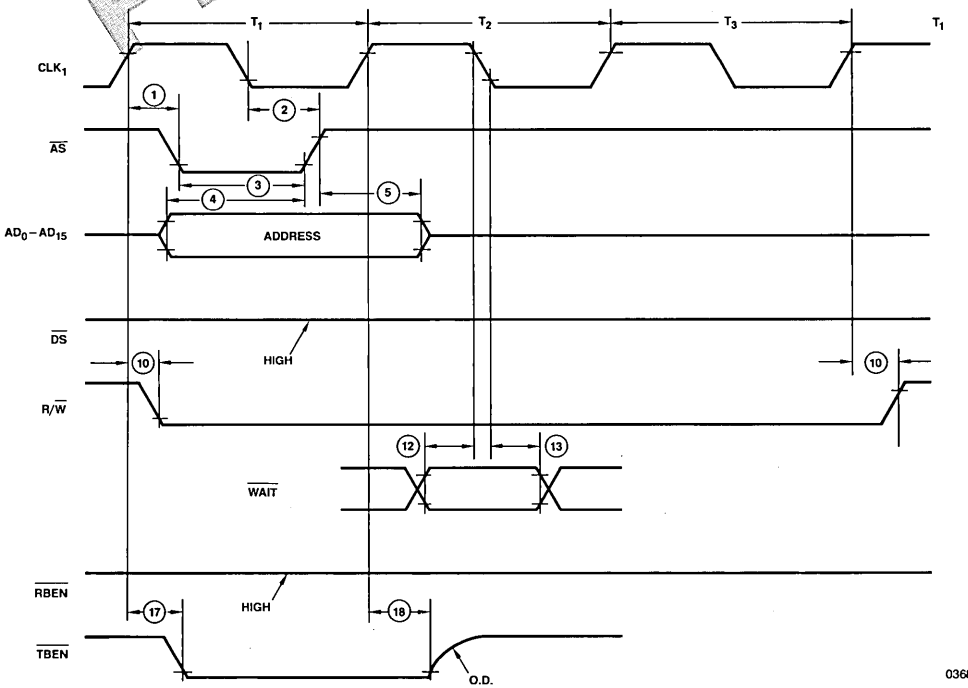


8052 BUS MASTER READ



03684B-17

8052 BUS MASTER WRITE



03684B-18

8052 BUS MASTER READ/WRITE

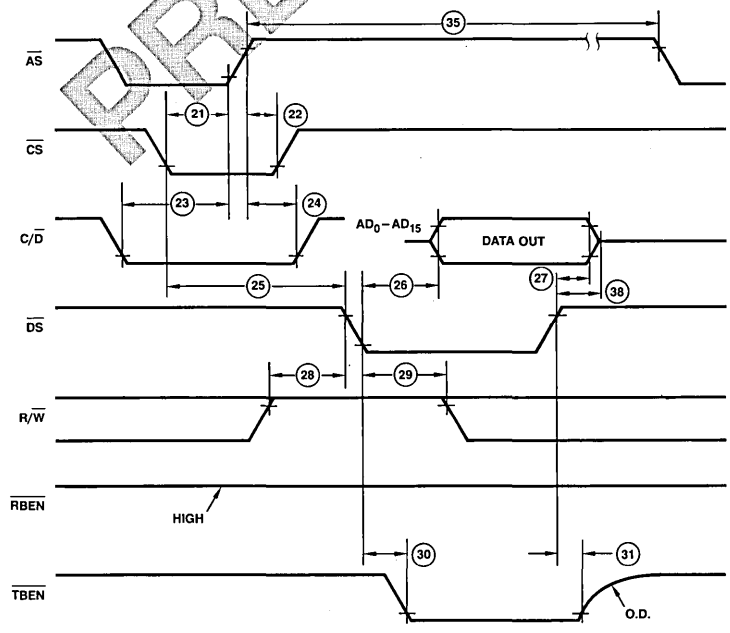
			6MHz		8MHz	
			Min	Max	Min	Max
1	t_{PHL}	$CLK_1\uparrow$ TO $\overline{AS}\downarrow$		55		45
2	t_{PLH}	$CLK\downarrow$ TO $\overline{AS}\uparrow$		55		45
3	t_{PW}	$\overline{AS}$ PULSE WIDTH	60		45	
4	t_S	ADDRESS VALID TO $\overline{AS}\uparrow$	40		30	
5	t_H	ADDRESS FROM $\overline{AS}\uparrow$	30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

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

8052 BUS SLAVE READ LATCHED

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

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

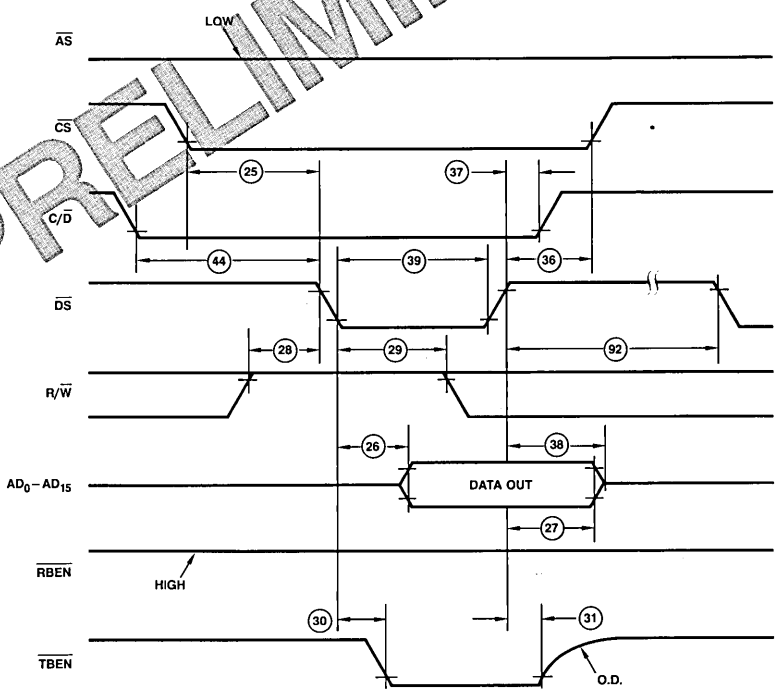


03684B-19

8052 BUS SLAVE READ UNLATCHED

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

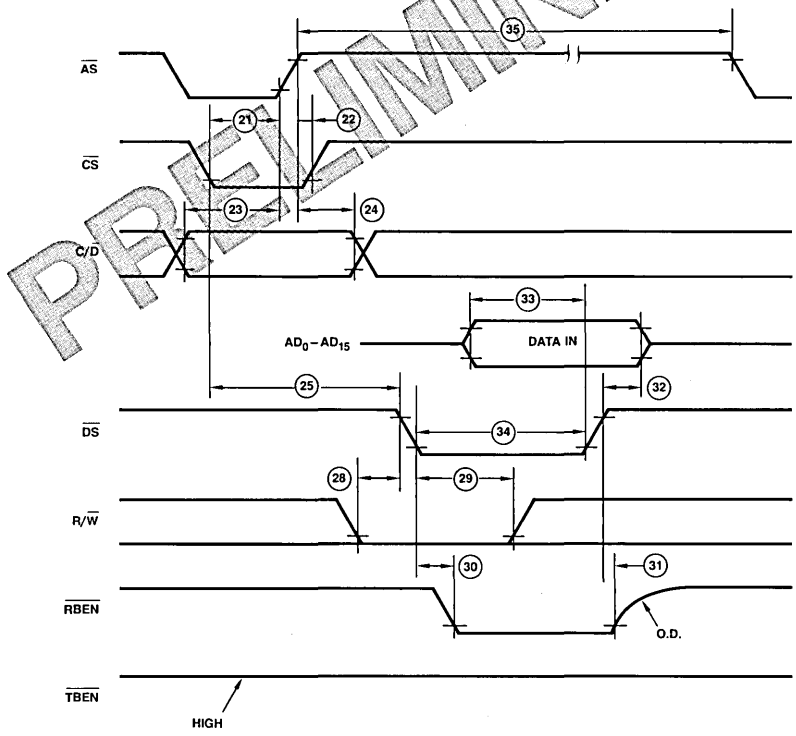
Note 1: t_{SRT}: (92) = 3 • (40) − (39)



03684B-20

8052 BUS SLAVE WRITE LATCHED

			6MHz		8MHz	
			Min	Max	Min	Max
32	t_H	DATA IN VALID FROM $\overline{DS}\uparrow$	0		0	
33	t_S	DATA IN VALID TO $\overline{DS}\uparrow$	90		80	
34	t_{PW}	$\overline{DS}$ PULSE WIDTH	135		100	

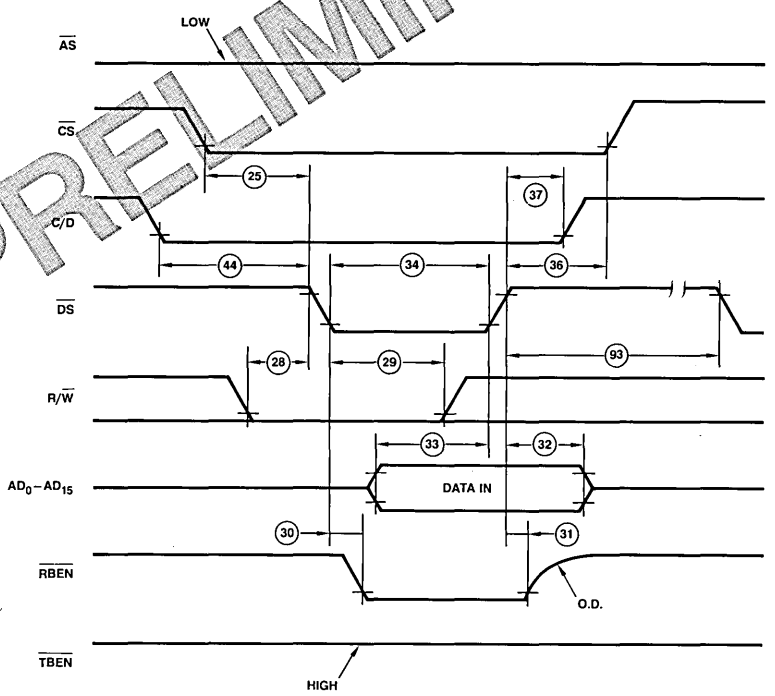


03684B-21

8052 BUS SLAVE WRITE UNLATCHED

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

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

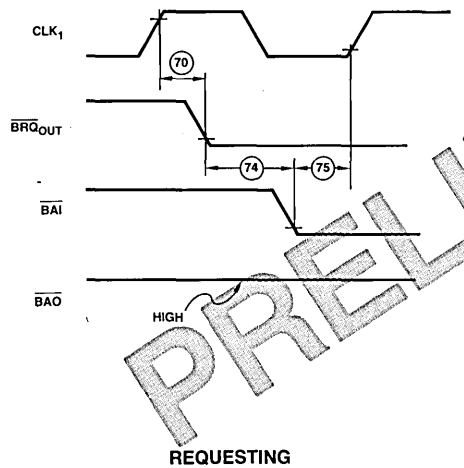


03684B-22

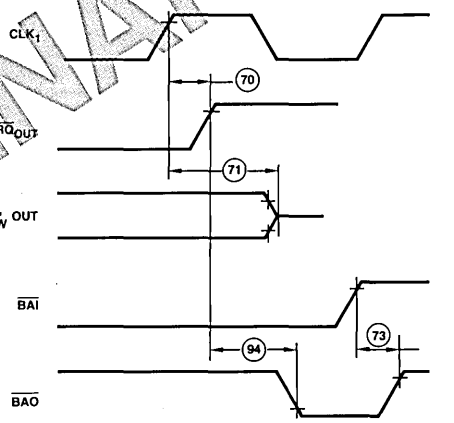
3052 BUS EXCHANGE

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

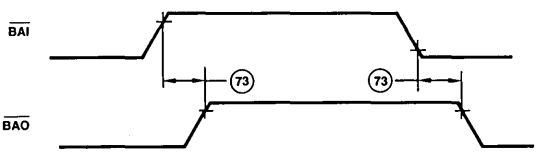
Note 1: This parameter for testing only.



REQUESTING



RELEASING

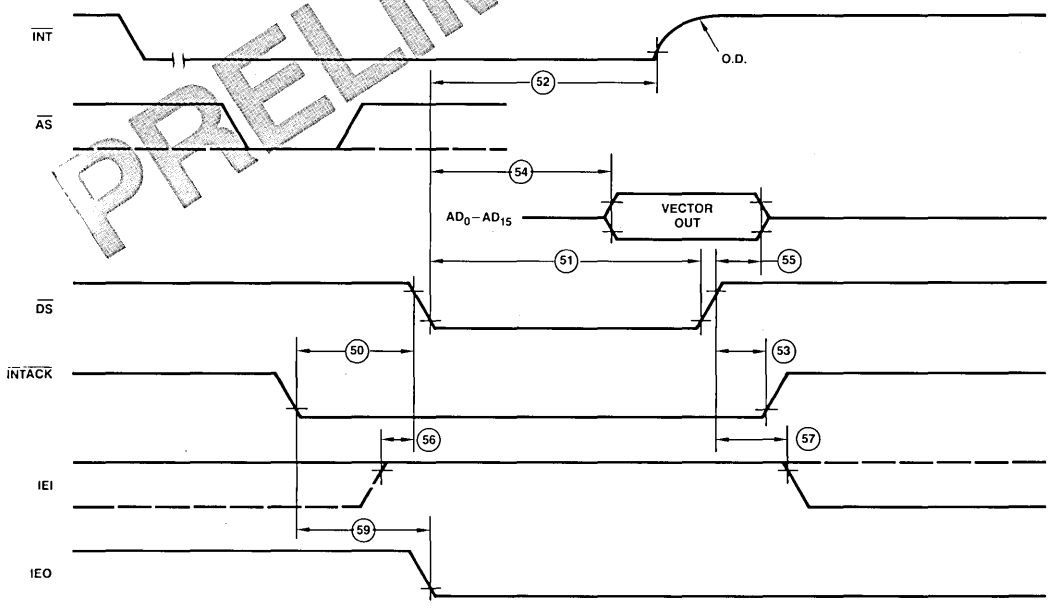


CHAIN DELAY

03684B-23

8052 INTERRUPT ACK TIMING – DEVICE ACKNOWLEDGED

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

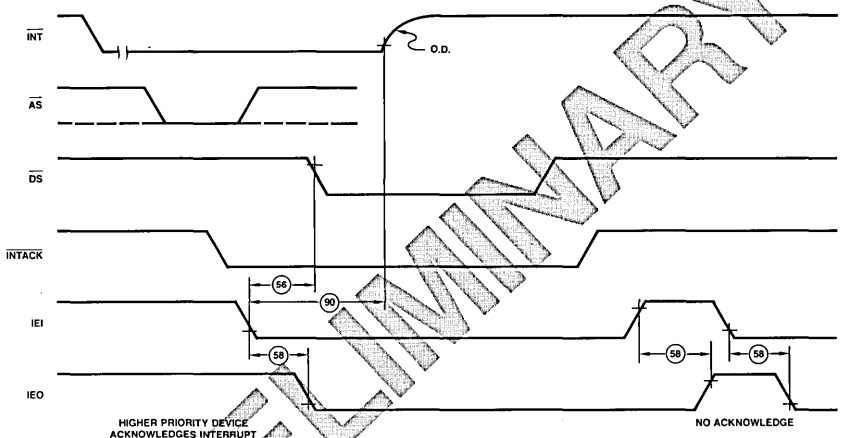


03684B-24

8052 INTERRUPT ACK TIMING – LOW PRIORITY

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

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



03684B-25

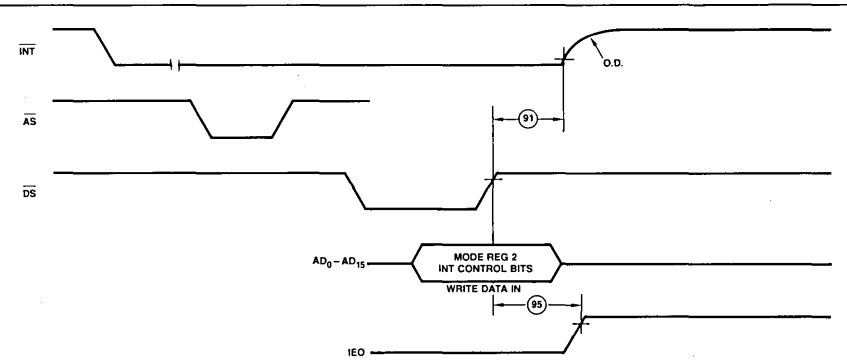
8052 NON-VECTORED INT TIMING

			6MHz		8MHz	
			Min	Max	Min	Max
91	t_D	$\overline{DS}\uparrow$ TO $\overline{INT}$ (Write) (Note 1)		90		80
95	t_D	$\overline{DS}\uparrow$ TO IEO $\uparrow$ (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 $\rightarrow$ 1 IUSV $\rightarrow$ 1
IES $\rightarrow$ 0 IES $\rightarrow$ 0
IPS $\rightarrow$ 0 IPV $\rightarrow$ 0

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



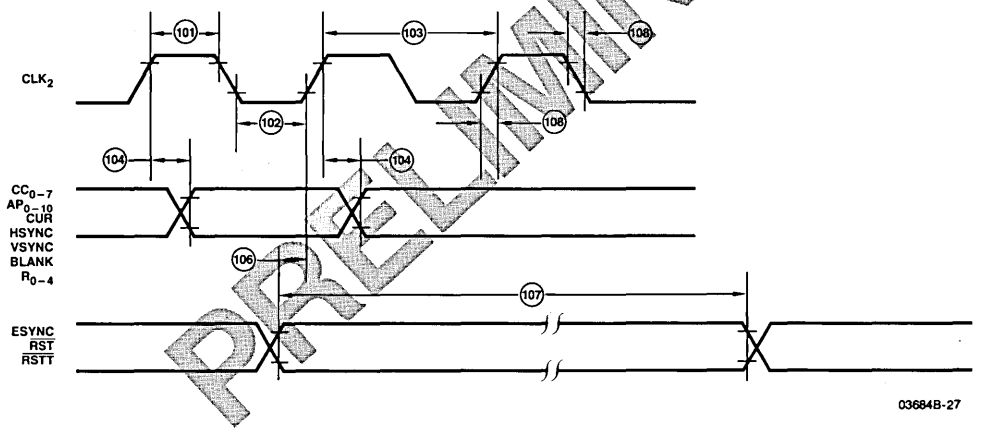
03684B-26

8052 VIDEO OUTPUTS AND SYNCHRONIZING INPUT TIMING

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

Notes: 1. Parameter 106 is specified for test purposes only.

2. Parameter 107 is for reset and reset only.
   T = CLK_2 period



ADVANCED MICRO DEVICES, INC.
901 Thompson Place
P.O. Box 3453
Sunnyvale,
California 94088
(408) 732-2400
TWX: 910-339-9280
TELEX: 34-6306
TOLL FREE
(800) 538-8450

05098-A

CD-B-2.5M-11/84-1