mach64
VGA Register Guide

Technical Reference Manuals

P/N: VGA-S00700-05

ATI Technologies Inc.
33 Commerce Valley Drive East
Thornhill, Ontario
Canada L3T 7N6

Developer Support:	905-882-2600 ext.6000
Offices:	905-882-2600
Fax:	905-882-2620
BBS:	905-764-9404

P/N: VGA-S00700-05

RELEASE 5.0

The information contained in this document has been carefully checked and is believed to be entirely reliable. No responsibility is assumed for inaccuracies. ATI reserves the right to make changes at any time to improve design and supply the best product possible.

Systems Publication Index

Technical Manuals

- ***mach64*** Graphics Controller Specifications (GCS-C015XX1-05)

- ***mach64*** Programmer's Guide (PRG-S00700-05)

- ***mach64*** VGA Register Guide (VGA-S00700-05)

- ***mach64*** Register Reference (RRG-S00700-05)

- ***mach64*** BIOS Kit (BIO-C012XX1-05)

Record of Revisions

Release	Date	Description of changes
01	93 Oct	PRELIMINARY
03	94 Feb	BETA RELEASE
04	94 May	BETA RELEASE
05	94 Oct	RELEASE

Contents

Chapter 1 About This Manual

Chapter 2 Introduction

Chapter 3 Overview

Chapter 1
Introduction

About This Manual

This manual is a register description for OEM designers and developers who wish to integrate ATI's *mach64* graphics controller accelerators in their hardware. This book covers the VGA portion of the *mach64*GX, CX, EX, and CT. It is also for programmers who wish to program directly to hardware registers, in order to optimize graphics performance. A working knowledge of the 80x86 family of PCs, Assembler, "C", and 2D graphics is assumed.

This manual consists of five chapters and an index.

- **Chapter 1, Introduction** Lists the *mach64* graphics controller's features, and describes the notation conventions used in this book.
- **Chapter 2, Programmer's Overview** Provides a programmer's overview of the *mach64* and its capabilities. Power-on setup registers and Peripheral Component Interconnect (PCI) configuration registers are described.
- **Chapter 3, VGA Controller** Describes the VGA controller, VGA display modes, and display memory organization.
- **Chapter 4, VGA-Compatible Registers** Discusses the VGA-compatible registers.
- **Chapter 5, VGA Register Extensions** Introduces ATI's VGA register extensions.

Features

- Low-cost, high-performance, single-chip graphics solution
- Suitable for board- or system-level implementations
- 100% register-level hardware compatible with the IBM VGA
- I/O Bus Types: ISA (8- and 16-bit), EISA (32-bit)
- Local Buses: 486, 386DX, 386SX, VESA VL-bus, and PCI
- Fully programmable direct memory interface
- Memory Types: *VRAM* 256Kx4, 256Kx16; *DRAM* 256Kx4, 256Kx16
- Memory Sizes: 512K, 1M, 2M, 4M, and 8M
- Colors/Resolutions:
  4 and 8 bpp (Bits Per Pixel) to 1280x1024
  16 bpp to 1280x1024
  24 bpp to 1280x1024
- High-speed, point-to-point line draw; coprocessor supports up to 32 bpp modes
- Supports memory-mapped registers
- Supports overscan
- Hardware cursor: 64x64x2
- High-speed polygon fill
- Hardware-assisted line and polygon pre-clipping
- Support for packed bitmap data transfers
- 32, 8-bit pixel color pattern registers
- 32, 1-bit monochrome pattern registers
- Enhanced bit block transfer (blit) operation to allow for better off-screen memory management
- Extended 16-entry data FIFO
- Improved FIFO status registers providing dramatic improvements in data throughput
- Card ID feature supports up to seven display adapters simultaneously in a system
- 0.7 micron CMOS VLSI technology
- 208-pin PQFP

Notation Conventions

Mnemonics are used throughout this manual in place of hardware register names. The naming convention for registers and/or bit fields is as follows:

- Register_Mnemonic
- Register_Mnemonic[Bit_Numbers or Field_Name]

The mnemonic for the Miscellaneous Output register is:

GENMO

The following examples describe the same entity in two different ways — as bits 2 and 3, and as the Clock_Select field of the GENMO register. Note that *square brackets* [ ] are used.

GENMO[3:2] or

GENMO[CLOCK_SELECT]

The convention for naming signals is similar to that used for naming hardware registers.

Signal_Name

When several signals of an identical function are described, the part of the signal name that differs may be shown in parentheses ( ). For example, the four Select signals — SEL0#, SEL1#, SEL2#, and SEL3# — may be represented as follows.

SEL(0:3)#

This page intentionally left blank.

Chapter 2
Programmer's Overview

Introduction

The *mach64* is a VLSI graphics controller chip consisting of a 64-bit GUI accelerator and a VGA-compatible graphics controller. The accelerator is also known as the **coprocessor** or the **draw engine**. This chip may be used to implement board- or system-level solutions supporting a range of I/O bus types, memory types, memory sizes, screen resolutions, and color depths. In addition, the chip supports a number of extended registers and enhancements.

VGA Controller

The built-in VGA controller can be disabled to allow the accelerator to co-exist with an alternate external VGA. This manual focuses on programming the VGA controller; for coprocessor operation, refer to the *mach64 Programmer's Guide.*

Configurable Memory Aperture

A configurable linear memory aperture is available for all modes, including VGA. In real mode, it may be configured as either a 64K aperture at A000h or two 32K apertures at A000h and A800h, depending on the graphics mode (as will be described later). In protected mode, it may be configured to 1M-paged or 8M-linear on any 4M or 8M boundry. The aperture is primarily used for increasing throughput on host-to-screen and screen-to-host data transfers.

Screen memory can be shared between the VGA and the GUI engine. *mach64*-aware applications may reconfigure the memory boundary (register) dynamically to give more or less memory to the VGA or the accelerator.

DACs

Since *mach64* boards may be equipped with a number of different DACs, a protocol has been established for accelerator mode switching by ROM calls. VGA mode switching is accomplished with the standard INT 10h interface.

The VGA and accelerator should be treated as two logically-separate entities. Bus type and memory type should be transparent to application programmers.

Hardware Cursor

A variable-size hardware cursor is available for all non-VGA modes. This cursor may be composed of two colors, transparent or complement.

mach64 VGA Variations

The *mach64* X-series of chips (GX, CX, EX) have full featured VGA controllers on-chip, compatible with the ATI28800 super VGA controller. The T-series (CT) chips have a basic VGA with VESA extensions on-chip. The T-series also include an internal DAC and an internal PLL.

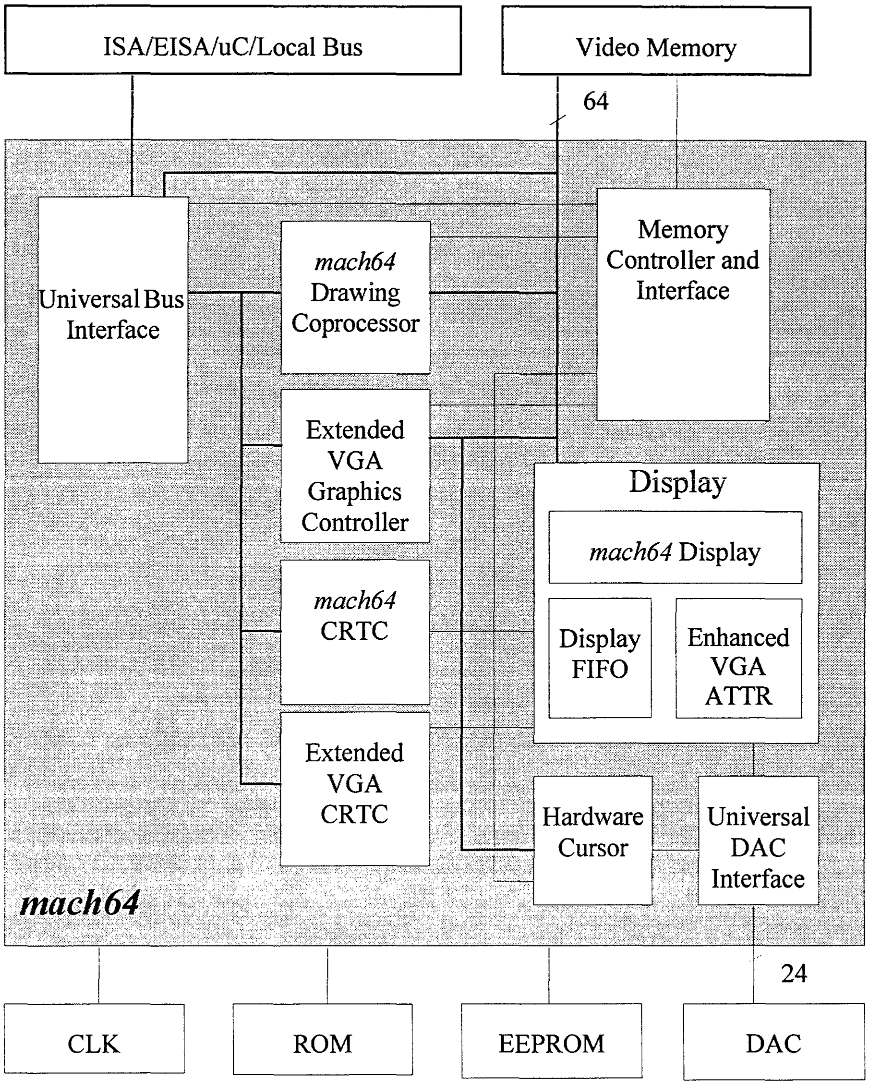


Figure 2-1. mach64GX, CX, and EX Drawing Coprocessor

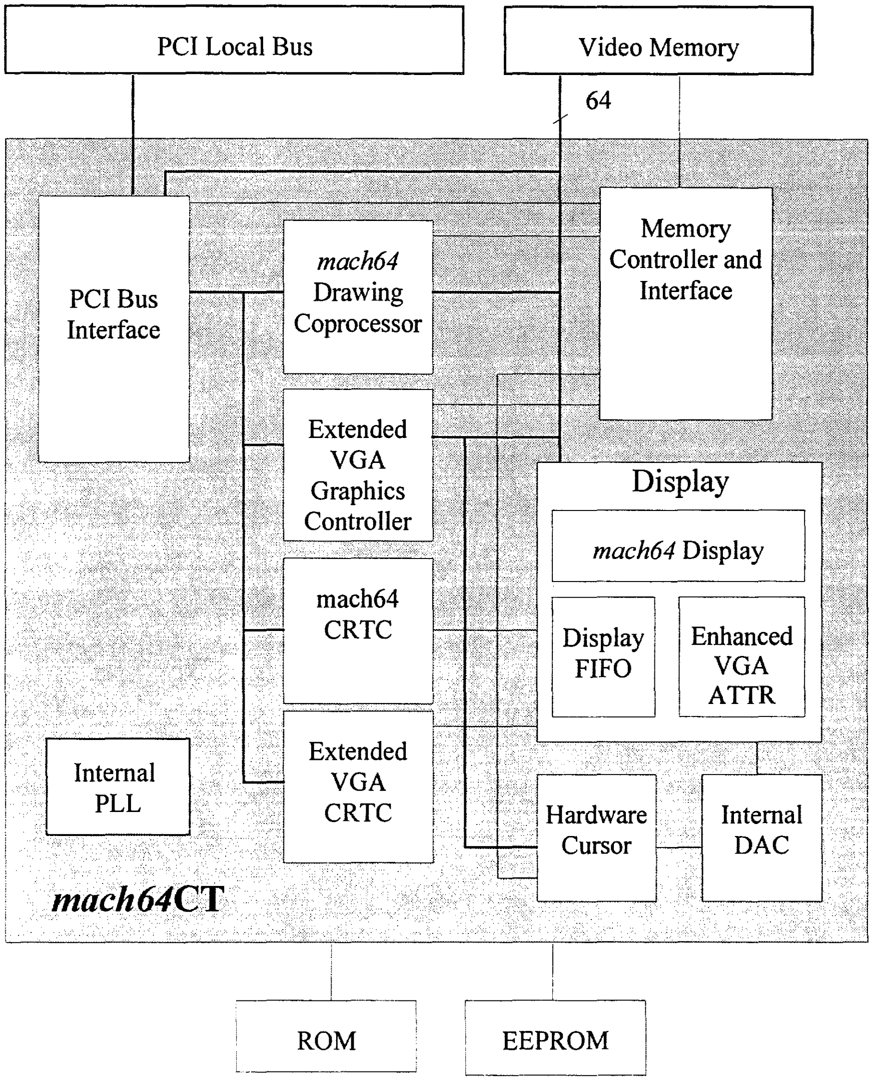


Figure 2-2. mach64CT Drawing Coprocessor

Power-on Setup Registers

EISA Bus POS Registers

These registers only exist in *mach64*'s which support EISA bus. POS register address for EISA are from 0zC80h to 0zC87h, inclusive. The input signal AEN must be "0" for these registers to be accessed. The "z" in the address represents the bus slot number.

	EISA Setup Mode ID Byte 1 Register							
	SETUP_ID1, EISA					I/O:0zC80 (R)		
	7	6	5	4	3	2	1	0
VGA	a							

DESCRIPTION	
a	SETUP_ID1(EISA) Setup Identification byte 1 is 06h.
Note: The "z" in the address is the bus slot number.	

	EISA Setup Mode ID Byte 2 Register							
	SETUP_ID2, EISA					I/O:0zC81 (R)		
	7	6	5	4	3	2	1	0
VGA	a							

DESCRIPTION	
a	SETUP_ID2(EISA) Setup Identification byte 2 is 89h.
Note: The "z" in the address is the bus slot number.	

EISA Setup Mode ID Byte 3 Register

	SETUP_ID3, EISA					I/O:0zC82 (R)		
	7	6	5	4	3	2	1	0
VGA	a							

DESCRIPTION	
a	SETUP_ID3(EISA) Setup Identification byte 3 is 44h.
Note: The "z" in the address is the bus slot number.	

EISA Setup Mode ID Byte 4 Register

	SETUP_ID4, EISA					I/O:0zC83 (R)		
	7	6	5	4	3	2	1	0
VGA	a							

DESCRIPTION	
a	SETUP_ID4(EISA) Setup Identification byte 4 is 00h.
Note: The "z" in the address is the bus slot number.	

EISA Setup Mode Option Select Register

	SETUP_OPT, EISA					I/O:0zC84 (R/W)		
	7	6	5	4	3	2	1	0
VGA								a

DESCRIPTION	
a	SETUP_OPT(EISA) **0** = Video adapter disabled — responds only toPOS registers. **1** = Video adapter enabled.
Note: The "z" in the address is the bus slot number.	

	EISA ROM Base Address Setup Register							
	ROM_SETUP, EISA					I/O:0zC85 (W)		
	7	6	5	4	3	2	1	0
VGA	b	a						

DESCRIPTION	
a	**SETUP_ROMADOR (EISA)** Setting of ROM base address corresponding to CPU addresses A17:A11, in 8514 only mode.
b	**SETUP_ROMEN (EISA)** **0** = BIOS ROM decodes disabled. **1** = BIOS ROM decodes enabled.
Note: The "z" in the address is the bus slot number.	

	EISA Setup Byte 1 Register							
	SETUP_1, EISA					I/O:0zC86 (W)		
	7	6	5	4	3	2	1	0
VGA	a							

DESCRIPTION	
a	**SETUP_1(EISA)** General purpose R/W register. The value of this register is read back at 52EEh.
Note: The "z" in the address is the bus slot number.	

	EISA Setup Byte 2 Register							
	SETUP_2, EISA					I/O:0zC87 (W)		
	7	6	5	4	3	2	1	0
VGA	a							

DESCRIPTION	
a	SETUP_2(EISA) General purpose R/W register. The value of this register is read back at 52EFh.
Note: The "z" in the address is the bus slot number.	

PCI Configuration Registers

These registers only exist in *mach64*'s which support the Intel Peripheral Component Interconnect (PCI) local bus. Please refer to the *PCI Local Bus Specification* for detailed descriptions of the PCI Configuration Space registers. A brief summary follows:

Address	Bits	R/W	Functions	Power-Up Default
0h	15:0	R	Vendor ID	1002
2h	15:0	R	Device ID	4758 (GX) or 4358 (CX) or 4558 (EX) or 4354 (CT)
4h	0	R/W	I/O Access Enable	0 (Disabled)
	1	R/W	Memory Access Enable	0 (Disabled)
	2	R	Bus Master Enable	0 Always (Disabled)
	3	R	Special Cycle Enable	0 Always (Disabled)
	4	R	Mem. Write & Invalidate Enable	0 Always (Disabled)
	5	R/W	Palette Snooping Enable	0 (Disabled)
	6	R	Parity Error Enable	0 Always (Disabled)
	7	R/W	Read Wait Cycle Control	1 (Add 1WS)
	8	R	SERR# Enable	0 Always (Disabled)
	15:9	-	*Reserved*	0
6h	8:0	-	*Reserved*	0
	10:9	R	DEVSEL Timing	1 (Medium)
	11	R/W	Signaled Target Abort	0 (No Target Abort)
	12	R	Received Target Abort	0 Always (Inactive)
	13	R	Received Master Abort	0 Always (Inactive)
	14	R	Signaled System Error	0 Always (Inactive)
	15	R	Parity Error Detected	0 Always (Inactive)

Address	Bits	R/W	Functions	Power-Up Default
8h	7:0	R	Revised ID	0
9h	7:0	R	Register-Level Prog. Interface	0
Ah	7:0	R	Sub-Class Code/Programmable Interface	00 (VGA Comp. Device) 80 (VGA Disabled)
Bh	7:0	R	Base-Class Code	03 (Display Controller)
Ch	7:0	R	Cache Line Size	0 (Not Used)
Dh	7:0	R	Latency Timer	0 (Not Used)
Eh	7:0	R	Header Type	0
Fh	7:0	R	BIST	0 (Not Used)
10h	21:0	R	Mem. Aperture Base Address	0
	31:22	R/W		
2F:14h	-	-	*Reserved*	0 Always
30h	0	R/W	BIOS ROM Enable	0
	7:1	R	*Reserved*	0 Always
31h	7:0	R	*Reserved*	0 Always
32h	15:0	R/W	BIOS ROM Base Address	0
3B:34h	-	-	*Reserved*	0
3Ch	7:0	R/W	Interrupt Line	0
3Dh	7:0	R	Interrupt Pin	1
3F:3Eh	-	-	*Reserved*	0
40h	1:0	R/W	I/O Base	0 = 2EC 1 = 1CC 2 = 1C8 3 = Reserved
FF:41h	-	-	*Reserved*	0

Chapter 3
VGA Controller

Overview

The VGA Controller registers are completely hardware-compatible with the registers of the IBM Video Graphics Array (VGA) adapter. They are also fully hardware-register-compatible with the IBM Color Graphics Adapter (CGA). In addition, the VGA Controller provides a full set of extended registers for enhanced features and performance.

The ATI controller is not only fast, it is also capable of displaying colors and resolutions beyond VGA in interlaced and non-interlaced modes. It supports 8-/16-bit ROM and I/O operations, 8-/16-bit video RAM data transfers, and zero-wait-state BIOS and video memory operations.

This chapter provides specific details on each mode — resolution and color support, horizontal/vertical sync and polarities, pixel clock rates, and interlacing (all modes are non-interlaced unless otherwise indicated).

The following VGA functional blocks are integrated within the VGA portion of the *mach64*:

- Address Decoder
- Sequencer Controller
- CRT Controller
- Graphics Controller
- Attribute Controller.

IBM-Compatible Modes

- *CGA*

40x25(16-color text)
80x25(16-color text)
320x200(2 sets of 4-color graphics)
640x200(2-color graphics)

- *VGA*

40x25(16/256K color text)
80x25(2/ or 16/256K color text)
320x200(4/,16/,or 256/256Kcolorgraphics)
640x200(2/, 4/, or 16/256K color graphics)
640x350(2/ or 16/256K color graphics)
640x480(2/ or 16/256K color graphics)

High-Resolution and Wide-Column Graphics/Text Modes

- ***Super VGA Graphics Modes***

640x480(16 or 256/256K)
800x600(16 or 256/256K)
1024x768(16 or 256/256K color)

- ***Text Modes***

132x25(2 or 16/64 color)
132x44(2 or 16 color)

VGA Memory Organization

The segment base address of video display buffers in the *mach64* graphics controller is configurable as required by the emulated video standard. The size of this buffer depends on the selected display mode — higher resolution requires a larger memory buffer.

Standard	Size	Memory Segment Base Address
VGA/EGA	128K	B8000 (Color Text) B0000 (Mono Text) A0000 (Graphics)
CGA	32K	B8000 (Color Text & Graphics)
MDA	32K	B0000 (Mono Text)

Memory organization affects how the video data of each supported display mode is written or read:

- A/N mode is alphanumeric (text).
- APA mode is All-Planes-Addressable (graphics).
- Modes 0 - 13 are 100% compatible with the modes available in the various IBM display adapters.

The display buffer can be stored as **page** or **map** memory.

- **Page memory access** is reading and writing the video data one map (byte) at a time.
- **Map memory access** is applying the same CPU address to multiple maps (parallel locations) of video memory for data on one pixel — each map contributes its portion of color/attribute specifications. The processing of map data uses a set of four, 8-bit latches that correspond to the maps. Mode 13 uses a type of page memory access called **packed pixel format**.

Memory Maps

In display mode 0, the buffer size is 2K, and the actual amount required for a full screen is 2000 bytes. The base address of the video buffer can start at either B8000, B8800, or B9000, etc. Mode 0 uses two bytes to describe each pixel, as follows — the first byte of the buffer is for character code, the second byte is for attribute, the third byte is for character code, and the fourth byte is for attribute, and so on.

When a display mode requires more than 64K of data to paint a full screen, as in display mode 62, the display buffer will map to multiple pages.

Functional Blocks

The *VGA Controller* has five major functional blocks:

- **Address Decoder**
- **Sequencer Controller**
- **CRT Controller**
- **Graphics Controller**
- **Attribute Controller.**

These are illustrated below, including the control and data paths. Also shown are the connections to the video memory and video DAC. Data and addresses enter via the CPU bus on the left side.

Video output in the form of RGB analog signals is available at the video DAC on the right side. The five functional blocks of this chip are described in detail in the following sections.

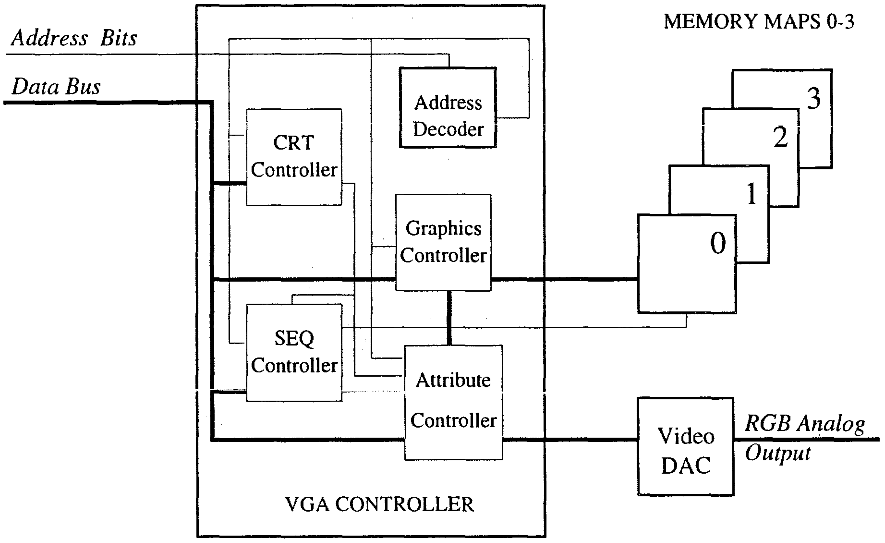


Figure 3-1. VGA Functional Block Diagram

Address Decoder

The address decoder is the top-level interface between the CPU and the Controller. Addresses and data from the input bus are decoded for each module. The starting address of video memory is programmable as VGA or CGA, for 100% adapter compatibility.

Sequencer Controller

The sequencer controller generates all timing signals for the video RAMs and all control signals for the other modules within the Controller. It prioritizes CRT and CPU accesses to the video memory.

Map mask register (plane mask register)

Video memory is protected from alteration by selectively masking out CPU writes to memory planes through the **map mask register** (also called the **plane mask register**).

CRT Controller

The CRT controller (CRTC) generates horizontal and vertical sync signals for the monitor interface, timings for cursor and underline attributes, timings for addressing the regenerative display buffer, and timings for refreshing the video RAM. CRT controller registers are user-programmable for controlling the display screen size, cursor type and position, character size, split screen, byte panning, and smooth scrolling.

Graphics Controller

The graphics controller handles data transfer between the CPU and video memory, with different types of pixel/data mappings for read and write operations. It is also the interface for latching display data from video memory to the attribute controller during the display cycles. Video memory data is sent to the attribute controller as 8-bit parallel data on alphanumeric modes, but as converted serial bit-plane data in graphics modes.

The graphics controller supports 8-bit and 16-bit bus operations. It also provides color comparators for applications such as color filling and boundary detection.

Attribute Controller

The attribute controller receives data from the graphics controller.

- In **text modes**, it generates video signals using the character generator and the attribute code.
- In **graphics modes**, it formats the video data into 1-, 2-, 4-, or 8-pixel streams, as required by the selected mode.

In either case, the video data is then passed through the internal 16/64 color palette registers and further processed to select a color value from the color palette register.

The output of the color register is then converted by a Digital to Analog Converter (DAC) to signals of the three primary colors, to drive a display device. The attribute controller also supports logic for blinking, underline, and horizontal pixel panning.

VGA Display Modes

The *mach64* chip supports alphanumeric or graphics display modes.

- Modes 0 to 6 are emulations of CGA modes. (Modes 0, 2, and 4 are identical to modes 1, 3, and 5, except that on the CGA adapter, modes 0, 2, and 4 have color burst turned off — color burst is not supported in the ATI68800 controller.)
- Modes with an asterisk (*) following their mode numbers are enhanced EGA modes, namely 0* to 3*.
- Modes with a plus symbol ($^+$) are enhanced VGA modes. They are 0^+, 1^+, 2^+, 3^+, and 7^+.

The standard and extended modes are fully described in the next section. Descriptions include character box size, screen resolution, and color/palettes.

VGA Alphanumeric Modes (A/N)

This section describes the supported A/N modes: IBM compatible modes 0, 1, 2, 3, 7, and ATI extended modes 23, 27, 33, and 37. Mode numbers are expressed as hexadecimal numbers.

- In A/N mode, the CPU transfers the character code for that mode into map 0, and attribute data into map 1. The CPU also transfers from ROM, the character patterns for that mode into map 2.
- Data from maps 0 and 1 are combined, one byte at a time, and read by the CRT controller.
- The CRTC then addresses the character generators in map 2. Dot patterns generated there are sent to the palette register, where a color value is assigned.
- According to this value, the DAC then produces the three RGB analog signals to drive the display.

Character byte and Attribute byte

The **character byte** describes the displayed character. The **attribute byte** describes color, intensity, etc. The four most significant bits (MSB) of the attribute byte define the background, and the other four bits describe the foreground (character).

Bit 3 of the attribute byte may also be used together with two 3-bit pointers in the Character Map Select register to produce a total of 512 characters and two separate character sets. If the video palette is changed, different colors will be generated. The attribute byte description is as follows:

Bit	"1"	"0"
7	Foreground Blinking	Background Highlighted
6	Background Red on	Background Red off
5	Background Green on	Background Green off
4	Background Blue on	Background Blue off
3	Foreground Highlighted* Enable SEQ03[5,3,2]	Foreground Normal Enable SEQ03[4,1,0]
2	Foreground Red on	Foreground Red off
1	Foreground Green on	Foreground Green off
	Foreground Blue on	Foreground Blue off

Bit 7							Bit 0
B	R	G	B	I	R	G	B

* The mode's default value is set by the BIOS. SEQ03[x,x] refers to bits xx of the SEQ03 register in the VGA graphics controller.

Each character on the screen is defined using two bytes of read/write memory. For example, on a 40x25 character page, a buffer memory of 2000 bytes is required. If each of the 40x25 characters has a resolution (box size) of 9x16 pixels, the page has a resolution of 360x400.

Video data in A/N modes is addressed one character at a time. See the following tables for descriptions of each supported mode.

Modes 0^+ and 1^+			
Standard	**Box Size**	**Char x Row**	**Colors**
VGA	9x16	40x25	16/256K

Modes 0 and 1			
Standard	**Box Size**	**Char x Row**	**Colors**
CGA	8x8	40x25	16

Modes 2^+ and 3^+			
Standard	**Box Size**	**Char x Row**	**Colors**
VGA	9x16	80x25	16/256K

Modes 2^+ and 3^+			
Standard	**Box Size**	**Char x Row**	**Colors**
CGA	8x8	80x25	16

Mode 7^+			
Standard	**Box Size**	**Char x Row**	**Colors**
VGA	9x16	80x25	2/256K

Mode 7			
Standard	**Box Size**	**Char x Row**	**Colors**
MDA	9x14	80x25	Mono

Modes 23, 27, 33, and 37			
Standard	**Box Size**	**Char x Row**	**Colors**
23	8x16	132x25	16/64
27	8x16	132x25	Mono
33	8x8	132x44	16
37	8x8	132x44	Mono

VGA Graphics Modes (APA)

This section describes the supported APA modes: IBM-compatible modes 4, 5, 6, D, E, F, 10, 11, 12, 13, and ATI extended modes 54, 55, 62, 63, and 64. Mode numbers are expressed in hexadecimal numbers.

Modes 4 and 5		
Standard	**Pels**	**Colors**
VGA	320x200	4/256K
CGA	320x200	4 (two sets)

Mode 6		
Standard	**Pels**	**Colors**
VGA	640x200	2/256K
CGA	640x200	2

Mode D		
Standard	**Pels**	**Colors**
VGA	320x200	16/256K

Mode E		
Standard	**Pels**	**Colors**
VGA	640x200	16/256K

Mode F		
Standard	**Pels**	**Colors**
VGA	640x350	2/256K

Mode 10		
Standard	**Pels**	**Colors**
VGA	640x350	16/256K

Mode 11		
Standard	**Pels**	**Colors**
VGA	640x480	2/256K

Mode 12		
Standard	**Pels**	**Colors**
VGA	640x480	16/256K

Mode 13		
Standard	**Pels**	**Colors**
VGA	320x200	256/256K (mono - 64 shades of gray)

Mode 54		
Standard	**Pels**	**Colors**
54	800x600	16/256K

Mode 55		
Standard	**Pels**	**Colors**
55	1024x768	16/256K

Mode 62		
Standard	**Pels**	**Colors**
62	640x480	256/256K

Mode 63		
Standard	**Pels**	**Colors**
63	800x600	256/256K

Mode 64		
Standard	**Pels**	**Colors**
64	1024x768	256/256K

Host Address Space/Host Window

VGA drawing operations are performed by the system processor, which reads data from and writes data to the on-screen display memory. To accomplish this, the display memory is mapped to a specific segment (or segments) of the host processor memory address space. This is sometimes referred to as the **host window** to display memory.

Standard memory organization of the 80x86 processor with 1 Megabyte of addressable memory is illustrated in table 3-1.

Table 3-1. PC Memory Map

Address	Content
F000:FFFF	
F000:0000	BIOS ROM
E000:0000	
	LAN, Tape Backup, EMS,...
CC00:0000	
C800:0000	XT Disk BIOS
C000:0000	VGA/EGA BIOS ROM
B800:0000	CGA Display Memory
B000:0000	MDA Display Memory
A000:0000	VGA/EGA Display Memory
	Transient Program Area (User memory)
	Resident part of COMMAND.COM
	Disk buffers, Installable Drivers, ...
	DOS Kernel
0000:0400	BIOS Data Area
0000:0000	Interrupt Vectors

The host window used by the VGA varies depending on the mode of operation. Table 3-2 contains the standard host windows, and sample modes using each window.

Table 3-2. VGA Host Windows

Content	Host Address Window
BIOS ROM	C000:0000h - C000:7FFFh
Display RAM Color Text (Mode 3)	B800:0000h - B800:7FFFh
Display RAM Monochrome Text (Mode 7)	B000:0000h - B000:7FFFh
Display RAM VGA Graphics (Modes F, 10,...)	A000:0000h - A000:FFFFh
Display RAM Extended Graphics	A000:0000h - A000:FFFFh
Display RAM Extended Graphics (128K addressable)	A000:0000h - B000:FFFFh

In text modes, which require relatively little data to be moved, a 32K space is used. In graphics modes, where much more data is required, a 64K space is used. When all four VGA color planes are used, the processor can access 256K of display memory (4 x 64K).

As the screen resolution and number of colors increase, the amount of display memory that must be accessed by the processor also increases. In some high resolution modes, more than 256K of display memory must be accessed.

A simple way to gain access to more display memory is to increase the size of the host memory space used by the VGA from 64K to 128K, using the memory address space from A000:0 to B000:FFFF. This standard VGA option, which is selected via the Miscellaneous Register of the Graphics Controller, is both convenient and efficient. However, it has the limitation in that it interferes with other co-resident display adapters such as MDA, CGA, or Hercules.

None of the standard modes on the *mach64* uses a 128K host window; 64K windows are used instead. An alternative way to access more than 64K is to use a display **memory mapping** mechanism, as discussed later in this chapter.

Packed vs. Planar Pixels

The color of each picture element (PEL or pixel) is determined by several bits in display memory (2, 4, or 8 bits, depending on the display mode). Two basic types of display memory organization are used to define pixels:

- **Planar organization (color planes)**. Most VGA graphics modes (Dh, Eh, Fh, 10h, 11h and 12h) use color planes.
- **Packed pixel organization**. CGA modes 4,5,6 and VGA mode 13 use packed pixels.

Planar Modes

In planar modes, each pixel on the screen corresponds to a set of four bits (one bit in each plane), as shown in figure 3-2.

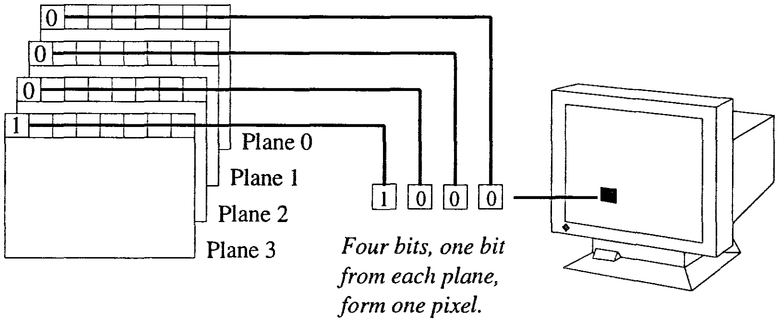


Figure 3-2. Planar Pixels

Packed Pixel Modes

For packed pixel modes, all bits of a pixel are packed into one or more bytes of display memory. In some modes, a byte holds more than one pixel: in 32K color modes, two bytes contain one pixel. In 256 color modes, a byte contains only one pixel, as shown in figure 3-3.

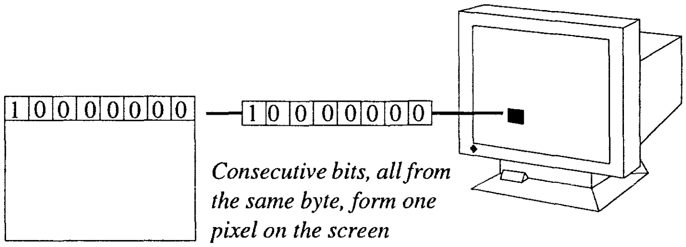

1 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0
Consecutive bits, all from the same byte, form one pixel on the screen

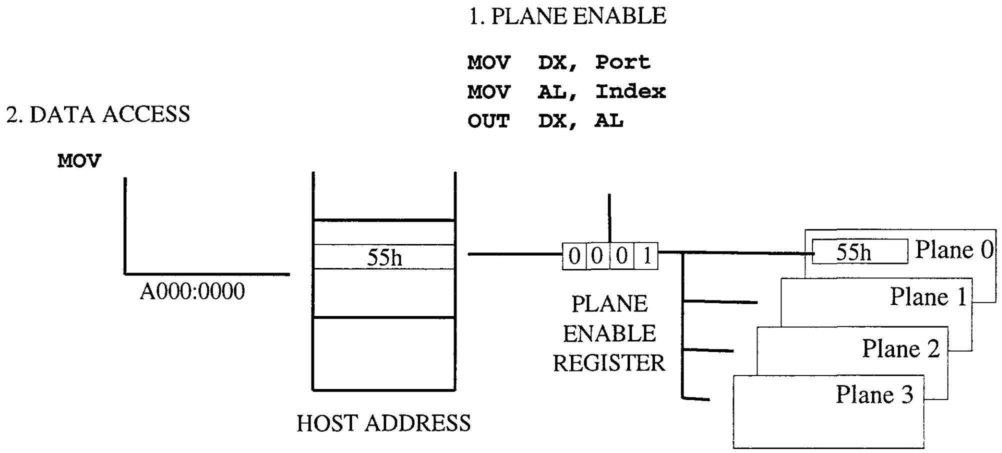
Figure 3-3. Packed Pixels

Display Memory Plane Selection

This display adapter can be configured with 256K, 512K, or 1M of display memory. Application programs access display memory via a 64K host window; to permit host access to all of the available display memory, the *mach64* contains two different memory mapping mechanisms that map the 64K host window to a selected 64K portion of display memory.

Planar organization of VGA memory

One memory mapping mechanism, which is standard on all VGA products, utilizes the planar organization of VGA memory. For a detailed explanation of the VGA memory planes see the text, *Programmer's Guide to the EGA/VGA* [1]. Planar memory mapping, illustrated in figure 3-4, allows access to 256K of display memory (four planes of 64K).



1. PLANE ENABLE

```
MOV  DX, Port
MOV  AL, Index
OUT  DX, AL
```

2. DATA ACCESS

```
MOV
```

Figure 3-4. Plane Selection

[1] Richard F. Ferraro, *Programmer's Guide to the EGA/VGA*, (Ontario:Allison-Welesy Publishing Co. Inc, 1990)

Display Memory Paging

To give the processor access to a greater amount of display memory, the *mach64* also contains a number of display memory paging mechanisms that map selected portions of the display memory to the host window. **Page** is defined as a section of display memory which is addressed by the CPU through A000:0000. Typically, the page size is 64K, but it in certain cases it may be 32K. It is similar in operation to EMS/LIM memory expansion boards.

The method for changing the page which is mapped to the host window is different for packed pixel and planar modes on the *mach64*. In packed pixel modes, data is read from and written to the screen at a selected page through small dual paged apertures. These apertures are 32K in size and are located at segment base addresses A000h and A800h. The read and write page of each aperture may be set independently. For a more detailed description of the small dual paged apertures and how to set their read and write pages, refer to the *mach64 Programmer's Guide*.

The method for changing the read or write page in planar modes varies depending on the type of *mach64* controller. For the *mach64GX*, *mach64CX*, and *mach64EX*, an application must first select the desired page by loading a page select register. Data within that memory page may then be accessed in the host window. Figure 3-3 illustrates this process. Each mode computes the page number and screen offset differently. Currently, the *mach64* supports up to 16 pages.

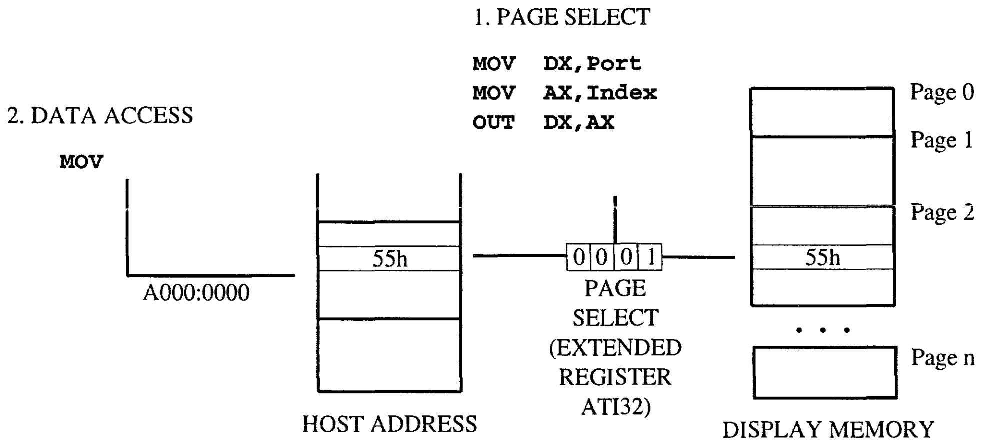


Figure 3-5. Page Selection

The page select register is part of a VGA extended register set contained within the *mach64GX*, *mach64CX*, and *mach64EX*. These registers are fully described in Chapter 5. To select a display memory page, set the appropriate bit in the extended register ATI32.

Since the *mach64*CT does not contain the set of VGA extended registers contained in the other series of *mach64* controllers, extended register ATI32 cannot be used to select the read or write page. For the *mach64CT,* the small dual paged apertures described earlier must be used to access the display memory in planar modes. See the *mach64 Programmer's Guide* for an explanation of how to modify the read or write page while using the small apertures in a planar mode.

Emulations

The *mach64* is compatible with software written for the Color Graphics Adapter (CGA). This compatibility is achieved by special registers that allow the *mach64* to emulate the CGA.

Extended Display Modes

In addition to total VGA compatibility, the *mach64* provides new enhanced display modes with higher resolutions and more colors than standard VGA display modes. The number of extended modes supported by the BIOS depends on the amount of memory available and the hardware revision.

- Modes that display 132 columns of text are useful for spreadsheets and similar applications.
- High-resolution graphics modes with 256 simultaneous color capability can be used to present full color photographic images with impressive fidelity.
- Modes that offer 16 colors at higher resolutions than the IBM VGA are popular for applications that involve fine visual details, such as CAD/CAM and desktop publishing.

Since the new modes were developed as extensions to the basic VGA, the programming models for these modes resemble the standard VGA modes. This means the effort required to become familiar with a new mode, and to modify existing software to utilize a new mode, is usually minimal. Table 3-3 lists the extended display modes supported by the *mach64.*

Table 3-3. Extended Display Modes

Mode	Type	Resolution	Colors	Memory Required
23h	Text	132 cols x 25 rows, 8x14 cell	16	256K
27h	Text	132 cols x 25 rows, 8x14 cell	mono	256K
33h	Text	132 cols x 44 rows, 8x8 cell	16	256K
37h	Text	132 cols x 44 rows, 8x8 cell	mono	256K
54h	Graphics	800 horizontal x 600 vertical	16	256K
55h	Graphics	1024 horizontal x 768 vertical	16 (planar)	512K
62h	Graphics	640 horizontal x 480 vertical	256	512K
63h	Graphics	800 horizontal x 600 vertical	256	512K
64h	Graphics	1024 horizontal x 768 vertical	256	1M

Mode Setting by BIOS Call

BIOS function 0, Mode Select, can be used to initialize any extended mode of the *mach64*. For example, to invoke mode 62(hex) the following code can be used:

```
MOV AH, 0        ;Load BIOS function code
MOV AL, 62h      ;Load mode number
INT 10H          ;Select mode
```

It is important to verify that the display being used is capable of supporting the colors and resolution of the selected display mode, and that the *mach64* has been properly configured for that display type. Otherwise, the mode select function may not initialize the mode properly.

Several of the extended modes require 512K or 1024K bytes of display memory to operate properly. To determine if a display mode is supported for a particular configuration, BIOS service 12h, subfunction 6 can be used.

This function returns a pointer in ES:BP to the standard VGA BIOS Video Parameter Table. BP is initially set to an invalid value before the function is called. Its value is checked after the function call to determine if the mode is supported. If BP is unchanged, then the mode is assumed to be unsupported. This procedure is illustrated in the following example:

Input Parameters:

AH = 12h

BL = 6

BH = 55h

AL = mode_number

SI = 0 (only used to determine if SI has changed)

BP = FFFFh (only used to determine if BP has changed)

Return Value:

ES:BP = pointer to parameter table

If register BP is unchanged, then the mode is not supported in the current configuration.

Example:

```
MOV     AH, 12h
MOV     BX, 5506h
MOV     AL, mode_number
MOV     BP, 0FFFFh
XOR     SI, SI
INT     10h
CMP     BP, 0FFFFh
JE      bad_mode
```

Summary of Display Modes

Extended text and graphics display modes of the *mach64* are summarized below.

Mode 23h - 132 x 25 Color Text

132-column text modes are useful for wide text displays such as spreadsheets. This mode uses the standard VGA 8x14 character set, and drives the display at a resolution of 1056 x 350 pixels. To support this mode, the display must be capable of displaying a horizontal resolution of 1056 pixels. While this exceeds the published specifications for most popular displays, acceptable results can still be achieved with many displays.

Mode 27h - 132 x 25 Monochrome Text

Mode 27h is the monochrome equivalent to mode 23h. It uses the same character set, and operates at the same resolution.

Mode 33h - 132 x 44 Color Text

A smaller character set is used in this text mode in order to display more information on the screen; the 8x8 character set used is not as readable as the 8x14 character set used in modes 23h and 27h. As with other 132-column modes, the screen operates at a resolution of 1056 x 350 pixels. Most displays will show acceptable results at this resolution.

Mode 37h - 132 x 44 Monochrome Text

Mode 37h is the monochrome equivalent to mode 33h.

Mode 54h - 800 x 600 16 Color Graphics

800x600 is the highest 16-color resolution that can be supported using only 256K bytes of display memory. It is also the highest 16-color resolution that can be supported without utilizing display memory paging. This resolution is also the upper limit of resolution on many popular displays.

Mode 55h - 1024 x 768 16 Color Graphics

Only the highest frequency VGA displays can operate in this mode, which is the highest resolution mode of the *mach64* VGA. 512K bytes of display memory are required. Display memory paging is used to make the full display memory available to the processor.

This resolution is becoming a standard for medium cost displays (640x480 being the previous standard), and has been added to many popular monitors to support the interlaced 8514/A adapter. The *mach64* VGA is capable of operating in this mode with interlaced as well as non-interlaced displays, and will automatically select the best one, based on the attached display (indicated by configuration settings). Programming in this mode can be accomplished through a 128K aperture located at address A000h (on ATI28800-5 chips and above).

Mode 62h - 640 x 480 256 Color Graphics

This resolution is popular because it equals the highest standard VGA resolution, but has added 256 color capability, and is supported by all low cost analog monitors. This mode requires 512K of display memory. Display memory paging is needed to make the larger display memory available to the processor.

Mode 63h - 800 x 600 256 Color Graphics

Mode 63h operates at the highest resolution that is available on the majority of multi-frequency (multi-scanning) monitors. Full color photographic images can be displayed with remarkable fidelity at this resolution. This mode requires 512K of display memory. Display memory paging is required to make the full display memory available to the processor.

Mode 64h - 1024 x 768 256 Color Graphics

Mode 64h operates at the highest resolution that is available on the majority of multi-frequency (multi-scanning) monitors. Full color photographic images can be displayed with remarkable fidelity at this resolution. This mode requires 1M bytes of display memory. Display memory paging is required to make the full display memory available to the processor.

Display Memory Organization

Drawing algorithms for a given display mode are largely dictated by the organization of display memory (the organization of a pixel in memory and the mapping of display memory to the screen) for that mode. This section provides detailed descriptions of display memory organization for all extended modes.

- For extended modes 54h, 55h, and 62h-64h, display memory organization is patterned after one of the standard IBM VGA modes.

Extended Text Modes

Modes 23h, 27h, 33h, and 37h

These modes utilize memory maps similar to those used in standard text modes (VGA modes 2, 3 and 7), except that the number of characters per line is increased from 80 to 132. Thus, the number of bytes used per text line increases from 160 to 264 (each character requires one ASCII character byte and one attribute byte). ASCII code is stored at even memory addresses, and attribute data is stored at odd memory addresses.

Table 3-4 shows the standard IBM color text attributes. In 16 color text mode, D7 is defined as foreground blinking, and D3 is the intensity bit. It allows one of sixteen colors for foreground and one of eight colors for the background. In addition, the character can be blinked.

Table 3-5 shows the standard IBM monochrome text attributes. D7 and D3 control the blinking and intensity respectively. Reverse video occurs only when bits 4-6 is one and bits 2-0 is zero. The organization of display memory for all extended text modes can be seen in figure 3-6. The character is stored in plane 0, and the attribute in plane 1. The ROM character patterns are transferred in plane 2 and addressed by the CRT Controller.

Table 3-4. Standard Color Text Attributes

Attribute	Standard Color	Attribute	Intensified Color
000	Black	1000	Gray
001	Blue	1001	Light Blue
010	Green	1010	Light Green
011	Cyan	1011	Light Cyan
100	Red	1100	Light Red
101	Magenta	1101	Light Magenta
110	Brown	1110	Yellow
111	Gray	1111	White

Table 3-5. Standard Monochrome Text Attributes

Attribute	Color
00000000	Blank
00000111	Normal
10000111	Blinking
00001111	Intensified
10001111	Blinking and Intensified
00000001	Underlined
10000001	Blinking and Underlined
00001001	Intensified and Underlined
10001001	Blinking, Intensified, and Underlined
01110000	Reverse Video
11110000	Reverse Video and Blinking

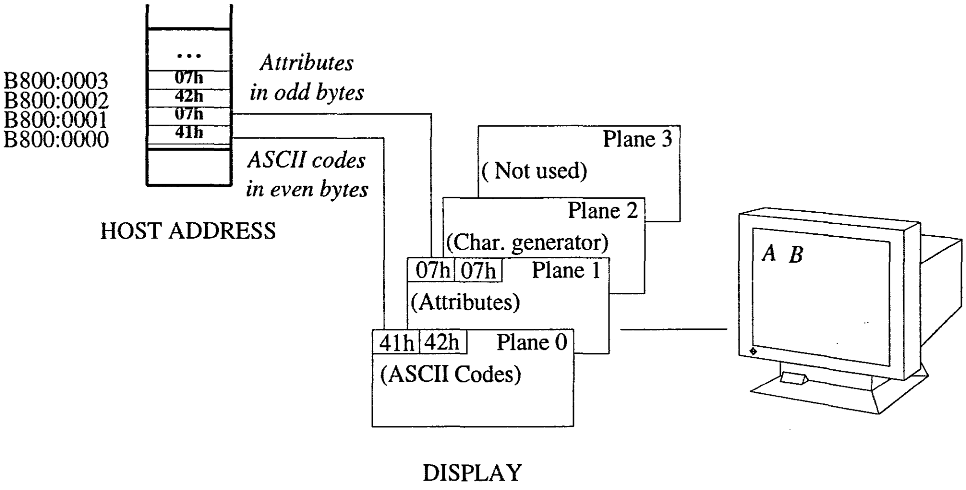


Figure 3-6. Display Memory Organization in Extended Text Modes

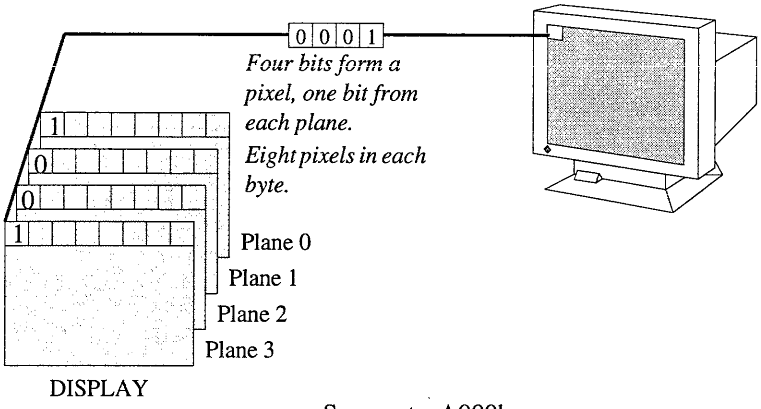
Extended Graphics Modes

Modes 54h - 800 x 600 (16 colors)

Memory organization for this mode resembles VGA mode 12h (640 x 480 16 color planar graphics), except that both the number of pixels per scan line and the number of scan lines increase. Display memory is organized as four color planes of 64K each. Each pixel consumes one bit position in each color plane (Plane Selection). The most significant bit in each byte (bit D7) corresponds to the leftmost pixel for that byte. One hundred consecutive bytes are used for each raster line.

For 600 lines, 60,000 bytes are required in each plane, which is less than the 64K that are addressable within the host window, so no display memory paging is required. Only 256K of display memory are required to support this mode.

Default colors are the same as the standard VGA colors (see table 3-4). To change colors in this mode, use the palette registers of the Attribute Controller.


0 0 0 1
Four bits form a pixel, one bit from each plane.
Eight pixels in each byte.
1
0
0
1
Plane 0
Plane 1
Plane 2
Plane 3
DISPLAY

Segment= A000h
Page = (64h x Y + X/8) / 10000h
Offset = (64h x Y + X/8) modulo 10000h
Bit Mask= 80h shr (X modulo 8)

Figure 3-7. Memory Organization - Mode 54h

Table 3-6. Standard VGA Color Palette - 16 Color Graphics

Index	Color
0000	Black
0001	Blue
0010	Green
0011	Cyan
0100	Red
0101	Magenta
0110	Brown
0111	White
1000	Dark Gray
1001	Light Blue
1010	Light Green
1011	Light Cyan
1100	Light Red
1101	Light Magenta
1110	Yellow
1111	Intensified White

Mode 55h - 1024 x 768 (16 colors)

Memory organization for this mode resembles VGA mode 12h (640 x 480 16 color planar graphics), except that both the number of pixels per scan line and the number of scan lines increase. Display memory is organized as two pages, with four color planes in each page, and 64K in each plane. Each pixel consumes one bit position in each color plane (see figure 3-2).

The most significant bit in each byte (bit D7) corresponds to the leftmost pixel for that byte. Each raster line uses 128 consecutive bytes. For 768 lines this requires 98,304 bytes in each plane, which is addressable in two, 64K pages in the host window. 512K of display memory are required to support this mode.

Default colors are the same as the standard VGA colors (see figure 3-4). To change colors in this mode, use the DAC registers.

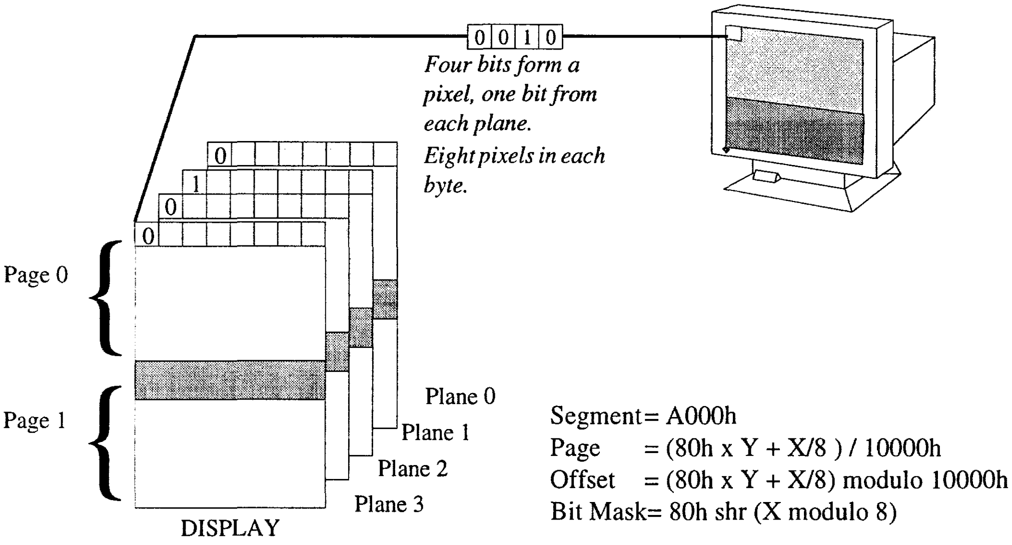
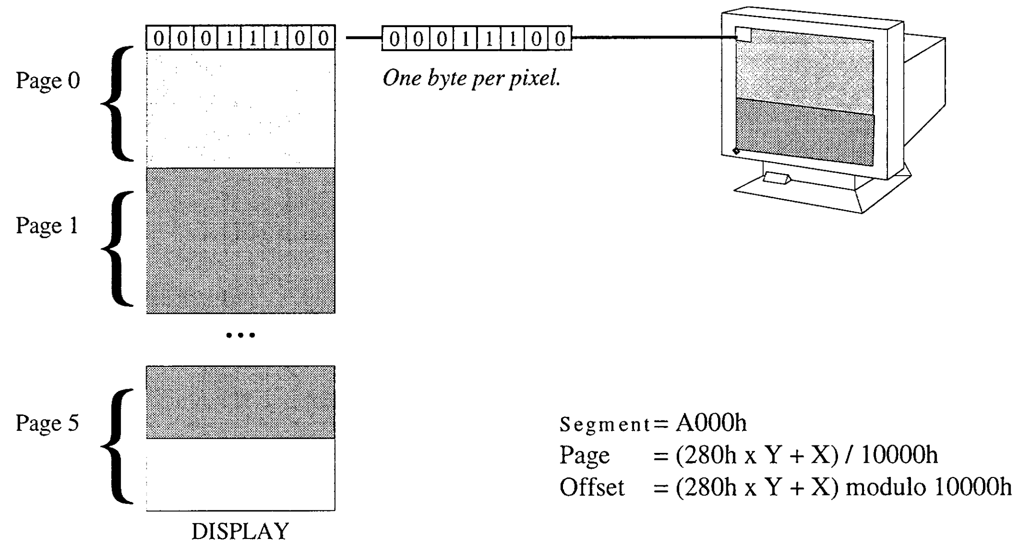

0 0 1 0
Four bits form a pixel, one bit from each plane.
Eight pixels in each byte.
0
1
0
0
Page 0
Page 1
Plane 0
Plane 1
Plane 2
Plane 3
DISPLAY
Segment= A000h
Page = (80h x Y + X/8) / 10000h
Offset = (80h x Y + X/8) modulo 10000h
Bit Mask= 80h shr (X modulo 8)

Figure 3-8. Display Memory Organization - Mode 55h

Mode 62h - 640 x 480 (256 colors)

Display memory organization for this mode resembles VGA mode 13h (320 x 200 256 color packed pixel graphics), except that the number of pixels per scan line is doubled and the number of scan lines is increased. Each pixel requires one byte of display memory. Each raster scan line uses 640 consecutive bytes. For 480 lines this requires 307.2K (five 64K pages) of display memory. The adapter requires 512K bytes of memory. The memory map for this mode can be seen in figure 3-9.

Default colors are the same as for mode 13h. To change the default colors, use the DAC registers; the palette registers in the Attribute Controller should not be modified.



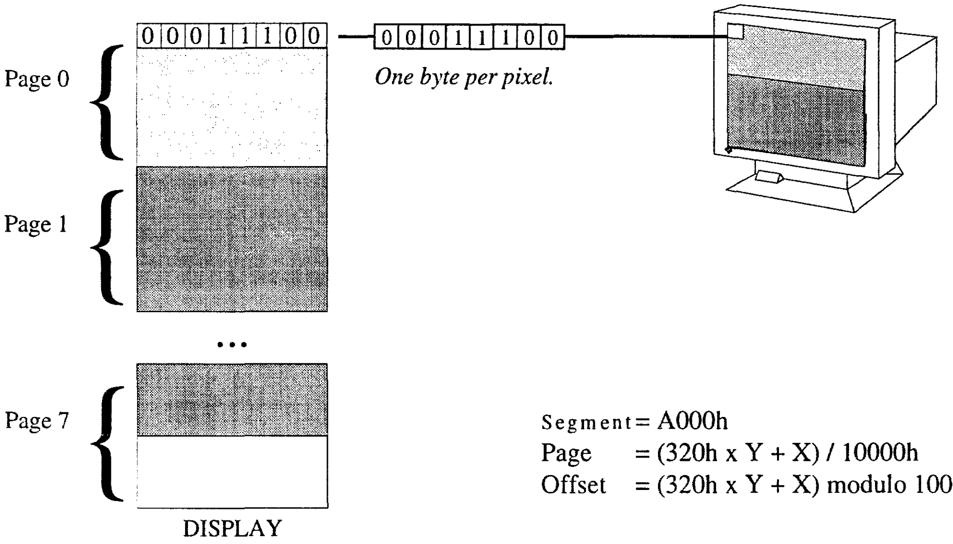
Segment = A000h
Page = (280h x Y + X) / 10000h
Offset = (280h x Y + X) modulo 10000h

Figure 3-9. Display Memory Organization - Mode 62h

Mode 63h - 800 x 600 (256 colors)

Display memory organization for this mode resembles VGA mode 13h (320 x 200 256 color packed pixel graphics), except that the number of pixels per scan line and number of scan lines are both increased. Each pixel requires one byte of display memory. Each raster scan line uses 800 consecutive bytes. For 600 lines this requires 480,000 bytes (eight 64K pages) of display memory. The adapter requires 512K bytes of memory. The memory map for this mode can be seen in figure 3-10.

Default colors are the same as for mode 13h. To change the default colors, use the DAC registers; the palette registers in the Attribute Controller should not be modified.



Segment = A000h
Page = (320h x Y + X) / 10000h
Offset = (320h x Y + X) modulo 10000h

Figure 3-10. Display Memory Organization - Mode 63h

Mode 64h - 1024 x 768 (256 colors)

Display memory organization for this mode resembles VGA mode 13h. Each pixel requires one byte of display memory and 1024 byte per scanline. The total display memory required for 768 lines is 786,432 bytes (12 64K pages). This mode works with an adapter of 1M memory only. The memory map for this mode is drawn in figure 3-11

Default colors are the same as for mode 13h. To change the default colors, use the DAC registers; the palette registers in the Attribute Controller should not be modified.

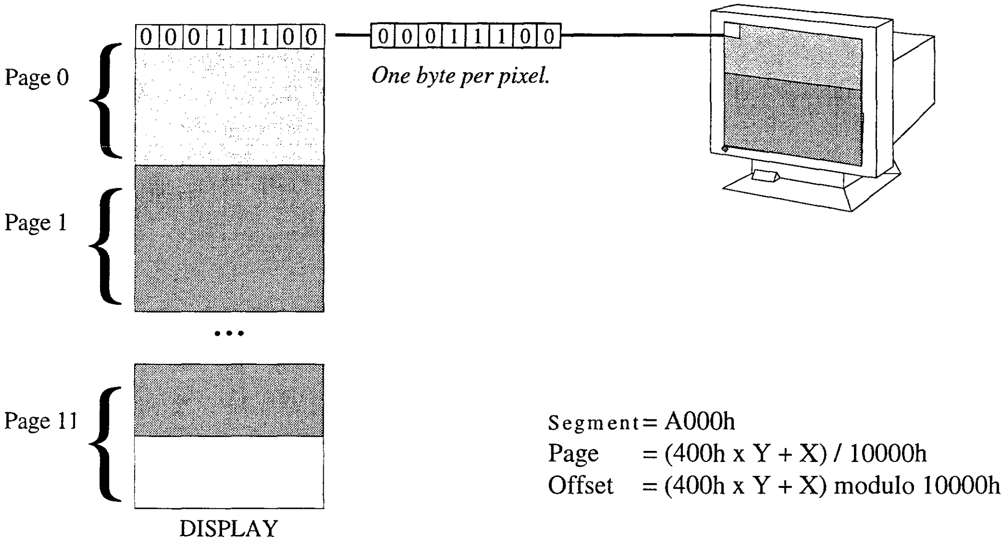


Figure 3-11. Display Memory Organization - Mode 64h

Table 3-7. Display Mode Specifications - PS/2 (Analog) Monitors

Mode	Box Size	PELs	Colors	Horiz. Sync (KHz)	Vertical Sync (Hz)	H/V Polarities	Dot Clock (MHz)
0+/VGA	9x16	360x400	16/256K	31.47	70	-/+	28.3
1+/VGA	9x16	360x400	16/256K	31.47	70	-/+	28.3
0/CGA	8x8	320x200	16	31.47	70	-/+	25.2
1/CGA	8x8	320x200	16	31.47	70	-/+	25.2
2+/VGA	9x16	720x400	16/256K	31.47	70	-/+	28.3
3+/VGA	9x16	720x400	16/256K	31.47	70	-/+	28.3
2/CGA	8x8	640x200	16	31.47	70	-/+	25.2
3/CGA	8x8	640x200	16	31.47	70	-/+	25.2
4/VGA	8x8	320x200	4/256K	31.47	70	-/+	25.2
4/CGA	8x8	320x200	4 (2 sets)	31.47	70	-/+	25.2
5/VGA	8x8	320x200	4/256K	31.47	70	-/+	25.2
5/CGA	8x8	320x200	4 (2 sets)	31.47	70	-/+	25.2
6/VGA	8x8	640x200	2/256K	31.47	70	-/+	25.2
6/CGA	8x8	640x200	2	31.47	70	-/+	25.2
7+/VGA	9x16	720x400	2/256K	31.47	70	-/+	28.3
7/MDA	9x14	720x350	Mono	31.47	70	+/-	28.3
D/VGA	8x8	320x200	16/256K	31.47	70	-/+	25.2
E/VGA	8x8	640x200	16/256K	31.47	70	-/+	25.2
F/VGA	8x14	640x350	2/256K	31.47	70	+/-	25.2
10/VGA	8x14	640x350	16/256K	31.47	70	+/-	25.2
11/VGA	8x16	640x480	2/256K	31.47	60	-/-	25.2
12/VGA	8x16	640x480	16/256K	31.47	60	-/-	25.2
12/VGA	8x16	640x480	16/256K	37.73	72.15	-/-	32
13/VGA	8x8	320x200	256/256K	31.47	70	-/+	25.2
23	8x16	1056x400	16/64	31.47	70	-/+	40
27	8x16	1056x400	Mono	31.47	70	-/+	40
33	8x8	1056x352	16	31.47	70	-/+	40
37	8x8	1056x352	Mono	31.47	70	-/+	40
55[1]	8x16	1024x768	16/256K	35.52	86	+/+	45

Table 3-7. Display Mode Specifications - PS/2 (Analog) Monitors

Mode	Box Size	PELs	Colors	Horiz. Sync (KHz)	Vertical Sync (Hz)	H/V Polarities	Dot Clock (MHz)
55	8x16	1024x768	16/256K	48.36	60	-/-	65
55	8x16	1024x768	16/256K	56.47	70	-/-	75
55	8x16	1024x768	16/256K	57.87	72	-/-	75
62	8x16	640x480	256/256K	31.47	60	-/-	25
62	8x16	640x480	256/256K	72.15	72	-/-	32
64[I]	8x16	1024x768	256/256K	35.52	87	+/+	45
64	8x16	1024x768	256/256K	48.36	60	+/+	65
64	8x16	1024x768	256/256K	56.48	70	-/-	75
64	8x16	1024x768	256/256K	57.87	72	-/-	75

Note:

I = Interlaced mode, 8514/A monitors only

Table 3-8. Display Mode Specifications - Multisync Monitors

Mode	Box Size	PELs	Colors	Hor. Sync (KHz	Vert Sync. (Hz)	H/V Polarities	Dot Clock (MHz)
0+/VGA	9x16	360x400	16/256K	31.47	70	-/+	28.3
1+/VGA	9x16	360x400	16/256K	31.47	70	-/+	28.3
0/CGA	8x8	320x200	16	31.47	70	-/+	25.2
1/CGA	8x8	320x200	16	31.47	70	-/+	25.2
2+/VGA	9x16	720x400	16/256K	31.47	70	-/+	28.3
3+/VGA	9x16	720x400	16/256K	31.47	70	-/+	28.3
2/CGA	8x8	640x200	16	31.47	70	-/+	25.2
3/CGA	8x8	640x200	16	31.47	70	-/+	25.2
4/VGA	8x8	320x200	4/256K	31.47	70	-/+	25.2
4/CGA	8x8	320x200	4 (2 sets)	31.47	70	-/+	25.2
5/VGA	8x8	320x200	4/256K	31.47	70	-/+	25.2
5/CGA	8x8	320x200	4 (2 sets)	31.47	70	-/+	25.2
6/VGA	8x8	640x200	2/256K	31.47	70	-/+	25.2
6/CGA	8x8	640x200	2	31.47	70	-/+	25.2
7+/VGA	9x16	720x400	2/256K	31.47	70	-/+	28.3
7/MDA	9x14	720x350	Mono	31.47	70	+/-	28.3
D/VGA	8x8	320x200	16/256K	31.47	70	-/+	25.2
E/VGA	8x8	640x200	16/256K	31.47	70	-/+	25.2
F/VGA	8x14	640x350	2/256K	31.47	70	+/-	25.2
10/VGA	8x14	640x350	16/256K	31.47	70	+/-	25.2
11/VGA	8x16	640x480	2/256K	31.47	60	-/-	25.2
12/VGA	8x16	640x480	16/256K	31.47	60	-/-	25.2
12/VGA	8x16	640x480	16/256K	37.73	72.15	-/-	32
13/VGA	8x8	320x200	256/256K	31.47	70	-/+	25.2
23	8x16	1056x400	16/64	31.47	70	-/+	40
27	8x16	1056x400	Mono	31.47	70	-/+	40
33	8x8	1056x352	16	31.47	70	-/+	40
37	8x8	1056x352	Mono	31.47	70	-/+	40
54	8x14	800x600	16/256K	35.16	56	-/-	36

Table 3-8. Display Mode Specifications - Multisync Monitors

Mode	Box Size	PELs	Colors	Hor. Sync (KHz	Vert Sync. (Hz)	H/V Polarities	Dot Clock (MHz)
54[V]	8x14	800x600	16/256K	35.16	56	+/+	36
54	8x14	800x600	16/256K	37.87	60	+/+	40
54	8x14	800x600	16/256K	44.19	70	+/+	45
55[I]	8x16	1024x768	16/256K	35.52	86	+/+	45
55	8x16	1024x768	16/256K	48.36	60	-/-	65
55	8x16	1024x768	16/256K	56.47	70	-/-	75
55	8x16	1024x768	16/256K	57.87	72	-/-	75
62	8x16	640x480	256/256K	31.47	60	-/-	25
62	8x16	640x480	256/256K	72.15	72	-/-	32
63	8x14	800x600	256/256K	35.16	56	-/-	36
63[V]	8x14	800x600	256/256K	35.16	56	+/+	36
63	8x14	800x600	256/256K	37.88	60	+/+	40
63	8x14	800x600	256/256K	48.04	72	+/+	50
64[I]	8x16	1024x768	256/256K	35.52	87	+/+	45
64	8x16	1024x768	256/256K	48.36	60	+/+	65
64	8x16	1024x768	256/256K	56.48	70	-/-	75
64	8x16	1024x768	256/256K	57.87	72	-/-	75

Notes:

I = Interlaced mode

V = Applicable to VESA-compatible monitors.

This page intentionally left blank.

Chapter 4
VGA-Compatible Registers

Overview

VGA registers in *mach64* controllers are fully hardware-compatible with registers in the IBM VGA video adapter. (See *Chapter 2, Programmer's Overview.*) In addition to the compatible registers, ATI's controllers also contain many extended-VGA registers that support higher resolutions, faster video modes, and enhanced features. These registers are described in *Chapter 5, VGA Register Extensions.*

Both VGA extended and compatible registers are listed in the *Index*. For convenience, they are also listed by I/O Port addresses on the following page. This chapter contains detailed descriptions of the compatible registers, arranged by function, as follows:

Register Classes	Page
General Registers (GENxx)	*4-5*
DAC Registers (DACxx)	*4-13*
Sequencer Registers (SEQxx)	*4-15*
CRT Controller Registers (CRTxx)	*4-20*
Graphics Controller Registers (GRAxx)	*4-40*
Attribute Controller Registers (ATTRxx)	*4-51*

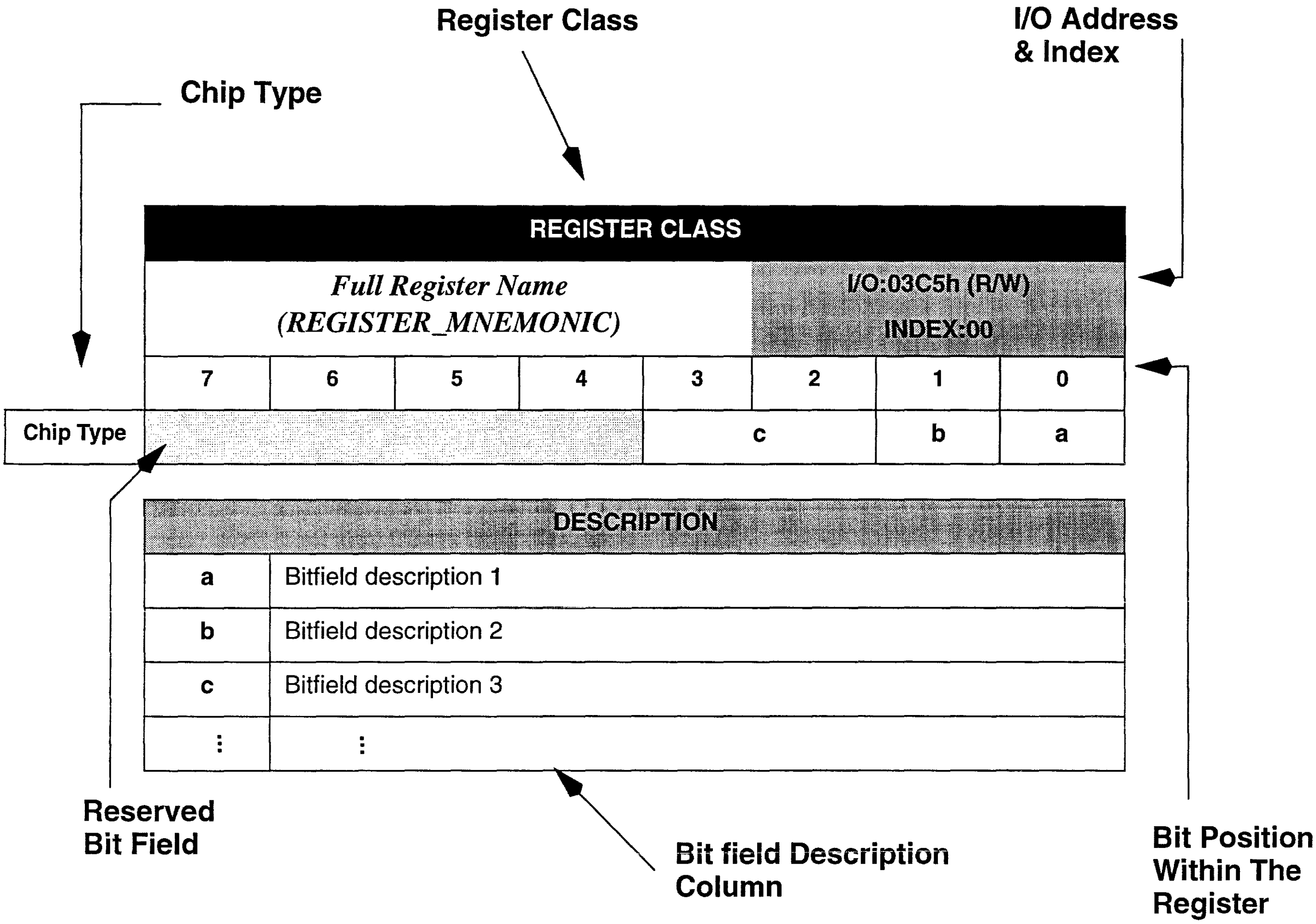
VGA Compatible Registers — By I/O Port

Port	Type	Mnemonic	Register Name	Page
0102	R/W	GENVS	VGA Sleep	*4-10*
03?4	R/W	CRTX	CRTC Index	*4-20*
03?5	R/W	CRT00	Horizontal Total	*4-20*
03?5	R/W	CRT01	Horizontal Display Enable	*4-21*
03?5	R/W	CRT02	Start Horizontal Blanking	*4-21*
03?5	R/W	CRT03	End Horizontal Blanking	*4-22*
03?5	R/W	CRT04	Start Horizontal Retrace	*4-23*
03?5	R/W	CRT05	End Horizontal Retrace	*4-23*
03?5	R/W	CRT06	Vertical Total	*4-24*
03?5	R/W	CRT07	CRTC Overflow	*4-24*
03?5	R/W	CRT08	Preset Row Scan	*4-26*
03?5	R/W	CRT09	Maximum Scan Line	*4-27*
03?5	R/W	CRT0A	Cursor Start	*4-28*
03?5	R/W	CRT0B	Cursor End	*4-29*
03?5	R/W	CRT0C	Start Address (High Byte)	*4-30*
03?5	R/W	CRT0D	Start Address (Low Byte)	*4-30*
03?5	R/W	CRT0E	Cursor Location (High Byte)	*4-31*
03?5	R/W	CRT0F	Cursor Location (Low Byte)	*4-31*
03?5	R/W	CRT10	Start Vertical Retrace	*4-32*
03?5	R/W	CRT11	End Vertical Retrace	*4-33*
03?5	R/W	CRT12	Vertical Display Enable End	*4-34*
03?5	R/W	CRT13	Offset	*4-34*
03?5	R/W	CRT14	Underline Location	*4-35*
03?5	R/W	CRT14	Start Vertical Blanking	*4-36*
03?4	R/W	CRT16	End Vertical Blanking	*4-36*
03?5	R/W	CRT17	CRT Mode	*4-37*
03?5	R/W	CRT18	Line Compare	*4-39*
03?8	R/W	GENMC	Mode Control Register (GENMC)	*4-8*
03CA	W	GENFC	Feature Control	*4-6*
03?A	R	GENS1	Input Status 1	*4-7*
03?B	R/W	GENLPC	Light Pen Clear	*4-12*
03B9	R	GENLPS	Light Pen Set (CGA) - Also see 03DC	*4-12*
03C0	W	ATTRX	Attribute Controller Index	*4-51*
03C0	W	ATTR(00:0F)	Palette (00 to 0F)	*4-52*
03C0	W	ATTR10	Mode Control	*4-53*
03C0	W	ATTR11	Overscan Color	*4-54*
03C0	W	ATTR12	Color Map Enable	*4-55*
03C0	W	ATTR13	Horizontal PEL Panning	*4-56*
03C0	W	ATTR14	Color Select	*4-57*
03C0	R	ATTRX	Attribute Controller Index	*4-51*
03C1	R	ATTR(00:0F)	Palette (00 to 0F)	*4-52*

Port	Type	Mnemonic	Register Name	Page
03C1	R	ATTR10	Mode Control	*4-53*
03C1	R	ATTR11	Overscan Color	*4-54*
03C1	R	ATTR12	Color Map Enable	*4-55*
03C1	R	ATTR13	Horizontal PEL Panning	*4-56*
03C1	R	ATTR14	Color Select	*4-57*
03C2	W	GENMO	Miscellaneous Output	*4-5*
03C2	R	GENSO	Input Status 0	*4-6*
03C3	R	GENENB	VGA Subsystem Enable (Board)	*4-9*
03C4	R/W	SEQX	Sequencer Index	*4-15*
03C5	R/W	SEQ00	Reset	*4-15*
03C5	R/W	SEQ01	Clocking Mode	*4-16*
03C5	R/W	SEQ02	Map Mask	*4-17*
03C5	R/W	SEQ03	Character Map Select	*4-18*
03C5	R/W	SEQ04	Memory Mode	*4-19*
03C6	R/W	DAC_MASK	DAC Mask Register	*4-13*
03C7	R/W	DAC_R_INDEX	DAC Read Current Color Index Register	*4-13*
03C8	R/W	DAC_W_INDEX	DAC Write Current Color Index Register	*4-14*
03C9	R/W	DAC_DATA	DAC Data Register	*4-13*
03CA	R	GENFC	Feature Control	*4-6*
03CC	R	GENMO	Miscellaneous Output	*4-5*
03CE	R/W	GRAX	Graphics Controller Index	*4-39*
03CF	R/W	GRA00	Set/Reset	*4-40*
03CF	R/W	GRA01	Enable Set/Reset	*4-41*
03CF	R/W	GRA02	Color Compare	*4-42*
03CF	R/W	GRA03	Data Rotate	*4-43*
03CF	R/W	GRA04	Read Map Select	*4-44*
03CF	R/W	GRA05	Graphics Mode	*4-45*
03CF	R/W	GRA06	Graphics Miscellaneous	*4-47*
03CF	R/W	GRA07	Color Don't Care	*4-48*
03CF	R/W	GRA08	Bit Mask	*4-49*
03D9	R/W	GENB	Border - Palette (CGA)	*4-11*
03DC	W	GENLPS	Light Pen Set (CGA) - Also see 03B9	*4-12*
46E8	W	GENENA	VGA Subsystem Enable (Add-On)	*4-9*

Register Legend

The table below describes the layout of all the Register Description Tables within Chapter 4 and 5.


Register Class
I/O Address & Index
Chip Type
Reserved Bit Field
Bit field Description Column
Bit Position Within The Register

REGISTER CLASS								
	Full Register Name (REGISTER_MNEMONIC)					I/O:03C5h (R/W) INDEX:00		
	7	6	5	4	3	2	1	0
Chip Type					c		b	a

DESCRIPTION	
a	Bitfield description 1
b	Bitfield description 2
c	Bitfield description 3
⋮	⋮

Register Class - Can be one of the register classes described on page 4-1.

Chip Type - Can be one of the following:

Chip Type	Description	Applicable Chip(s)
CGA	CGA compatible register	CGA, GX, CX, EX
VGA	VGA compatible register	VGA, GX, CX, EX, CT
ATI VGA	VGA register extension on ATI specific chips	GX, CX, EX

I/O Address & Index - Indicates the read and write address of the register. The index is also shown if the register is accessible indirectly

GENERAL REGISTERS

Miscellaneous Output Register (GENMO)

I/O:3CCh (R)
I/O:3C2h (W)

	7	6	5	4	3	2	1	0
VGA	e		d	▨	c		b	a

DESCRIPTION	
a	I/O Address Select **0** = Addressing for monochrome emulation. **1** = Addressing for color/graphic emulation.
b	Video RAM Enable **0** = Disables CPU access to video RAM. **1** = Enables CPU access to video RAM.
c	Clock Select **00** =25.1744 MHz (640 PELs). **01** =28.3212 MHz (720 PELs). **10** = *Reserved.* **11** = *Reserved.* To change these two bits, SEQ0[0:1] must first be set to zero.
d	Even/Odd Mode Page Select This bit is used in Even/Odd display modes (A/N modes: 0, 1, 2, 3, and 7). This bit is ignored when bit GRA06[1] or SEQ4[3] is enabled. **0** = Selects even (low) video memory locations. **1** = Selects odd (high) video memory locations.
e	Dual purpose bits used to select screen size and retrace sync polarity. (x=Bit not used for selection) Screen Size: **00** = *Reserved.* **01** =Screen size is 400 lines. **10** =Screen size is 350 lines. **11** =Screen size is 480 lines. Sync Polarity: **x0** =H Retrace pulse is active high. **x1** =H Retrace pulse is active low. **0x** =V Retrace pulse is active high. **1x** =V Retrace pulse is active low.

Note: In VGA mode, this register controls I/O port and video buffer addressing, and selects the dot clock frequency.

	GENERAL REGISTERS							
	Feature Control Register (GENFC)					I/O:3CAH (R) I/O:3?AH (W)		
	7	6	5	4	3	2	1	0
VGA					a			

DESCRIPTION	
a	Vertical Sync Select: **0** = Normal vertical sync. **1** = Sync is `vertical sync' ORed `vertical display enable'.
Note: **?** = B when GENMO[0]=0 (Monochrome emulation). **?** = D when GENMO[0]=1 (Color/Graphics emulation).	

	GENERAL REGISTERS							
	Input Status 0 Register (GENS0)					I/O:3C2h (R)		
	7	6	5	4	3	2	1	0
VGA	b			a				

DESCRIPTION	
a	Switch Sense: **0** = Output state of the DAC Lookup Table Comparators is inactive. **1** = Output state of the DAC Lookup Table Comparators is active.
b	CRT Interrupt: **0** = Vertical retrace interrupt is cleared. **1** = Vertical retrace interrupt is pending.

	GENERAL REGISTERS							
	Input Status 1 Register (GENS1)				**I/O:3?Ah (R)**			
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VGA			**c**		**b**			**a**

DESCRIPTION	
a	Display Enable: **0** = Enables display of video data. **1** = Disables display of video data.
b	Vertical Retrace Status: **0** = V Retrace pulse is inactive. **1** = V Retrace pulse is in progress.
c	Diagnostic Bits 0, 1 respectively: These two bits are connected to two of the eight color outputs (P7:P0) of the attribute controller. Connections are controlled by ATTR12[5,4] as follows: **00** =P2, P0 **01** =P5, P4 **10** =P3, P1 **11** =P7, P6

Notes:

1. **?** = B when GENMO[0]=0 (Monochrome emulation).
   ? = D when GENMO[0]=1 (Color/Graphics emulation).
2. Bits 0 and 3 can be used to synchronize the video buffer updates with the screen refresh cycles to minimize interference with the displayed image.

	GENERAL REGISTERS							
	Mode Control Register (GENMC)					I/O:3D8h (R/W)		
	7	6	5	4	3	2	1	0
CGA			**f**	**e**	**d**	**c**	**b**	**a**

DESCRIPTION	
a	**0** = 40x25 Text. **1** = 80x25 Text.
b	**0** = Text mode. **1** = Graphics mode.
c	**0** = Color mode **1** = B/W mode
d	**1** = Enables video
e	**1** = Enables 640x200 B/W mode
f	**0** = Keeps background intensity attribute bit. **1** = Changes background intensity attribute bit.

	GENERAL REGISTERS							
	Video Subsystem Enable (Add-On) Register (GENENA)					I/O:46E8h (W)		
	7	6	5	4	3	2	1	0
VGA				b	a			

DESCRIPTION	
a	VGA Enable: **0** = Puts VGA video subsystem into sleep mode, during which, the VGA video subsystem only responds to memory read operations to the BIOS ROM, and I/O writes to register 102h. All other I/O or video memory read/write operations are suspended. **1** = Enables I/O and memory address decoding of the VGA video subsystem, if GENVS[0] is also a logical one.
b	GENVS[0] Enable: **0** = Disables I/O write to GENVS (0102). **1** = Enables I/O write to GENVS (0102).

	GENERAL REGISTERS							
	Video Subsystem Enable(Board) Register (GENENB)					I/O:3C3h (R)		
	7	6	5	4	3	2	1	0
VGA								a

DESCRIPTION	
a	VGA Enable: Read back status of GENVS[0] (0102).

	GENERAL REGISTERS							
	VGA Sleep Register (GENVS)					I/O:102h (R/W)		
	7	6	5	4	3	2	1	0
VGA								a

DESCRIPTION	
a	VGA Sleep: **0** = Disables VGA video subsystem (controller). The VGA video subsystem only responds to memory read operations to the BIOS ROM. All other I/O or memory read/write operations are suspended. **1** = Enables VGA video subsystem.

Notes:

1. Writes to this register are controlled by GENENA[4].
2. Example — to enable the VGA:

```
MOV  DX, 46E8
MOV  AL, 10
OUT  DX, AL
MOV  DX, 102
MOV  AL, 1
OUT  DX, AL
MOV  DX, 46E8
MOV  AL, 8
OUT  DX, AL
```

	GENERAL REGISTERS							
	Border (Palette) Register (GENB)				I/O:3D9h (R/W)			
	7	6	5	4	3	2	1	0
CGA			f	e	d	c	b	a

	DESCRIPTION
a	Selects a blue border in 40x25, 80x25 text modes; 16-Color 320x200, 16-Color 640x200 graphics modes. Selects a blue background in 4-Color 320x200, 4-Color 640x200 graphics modes. Selects blue as foreground in 640x200 B/W mode.
b	Selects a green border in 40x25, 80x25 text modes; 16-Color 320x200, 16-Color 640x200 graphics modes. Selects a green background in 4-Color 320x200, 4-Color 640x200 graphics modes. Selects green as foreground in 640x200 B/W mode.
c	Selects a red border in 40x25, 80x25 text modes; 16-Color 320x200, 16-Color 640x200 graphics modes. Selects a red background in 4-Color 320x200, 4-Color 640x200 graphics modes. Selects red as foreground in 640x200 B/W mode.
d	Selects an intensified border color in 40x25, 80x25 text modes; 16-Color 320x200, 16-Color 640x200 graphics modes. Selects an intensified background color in 4-Color 320x200, 4-Color 640x200 graphics modes. Selects an intensified foreground color in 640x200 B/W mode.
e	Selects an intensified set of foreground colors in 4-Color 320x200 or 4-Color 640x200 graphics mode.
f	Selects an active set of colors in 4-Color 320x200 or 4-Color 640x200 graphics mode.
Note: This register is used in CGA emulation mode.	

GENERAL REGISTERS								
	Light Pen Set Register (GENLPS)					I/O:3DCh (W)		
	7	6	5	4	3	2	1	0
CGA	a							

DESCRIPTION	
a	Reading or writing at this I/O location sets the Light Pen Set register.
Note: This register is used in CGA emulation mode.	

GENERAL REGISTERS								
	Light Pen Clear Register (GENLPC)					I/O:3DBh (R/W)		
	7	6	5	4	3	2	1	0
CGA	a							

DESCRIPTION	
a	Reading or writing at this I/O location clears the Light Pen Clear register.
Note: This register is used in CGA emulation modes.	

	DAC REGISTERS							
	DAC Data Register (DAC_DATA)					I/O:03C9h (R/W)		
	7	6	5	4	3	2	1	0
VGA	a							

DESCRIPTION	
a	**DAC_DATA** DAC data

	DAC REGISTERS							
	DAC Mask Register (DAC_MASK)					I/O:03C6h (R/W)		
	7	6	5	4	3	2	1	0
VGA	a							

DESCRIPTION	
a	**DAC_MASK** Participating bit positions in the mask for DAC lookup are set to one.

	DAC REGISTERS							
	DAC Read Current Color Index Register (DAC_R_INDEX)					I/O:03C7h (R/W)		
	7	6	5	4	3	2	1	0
VGA	a							

DESCRIPTION	
a	**DAC_R_INDEX** The current read index for a DAC operation — increments after every third read of DAC_DATA (03C9h). Also see DAC_W_INDEX (03C8h).

	DAC REGISTERS							
	DAC Read Current Color Index Register (DAC_R_INDEX)					I/O:03C7h (R/W)		
	7	6	5	4	3	2	1	0
VGA	a							

DESCRIPTION
Note: Only bit 0 of this register is readable. Writing the DAC at 03C8h results in a read-back value of 0. Writing the DAC at 03C7h results in a read-back value of 1.

	DAC REGISTERS							
	DAC Write Current Color Index Register (DAC_W_INDEX)					I/O:03C8h (R/W)		
	7	6	5	4	3	2	1	0
VGA	a							

DESCRIPTION	
a	**DAC_W_INDEX** Current write index for a DAC operation. Also see DAC_R_INDEX (03C7h).

SEQUENCER REGISTERS

Sequencer Index Register (SEQX)

I/O:3C4h (R/W)

	7	6	5	4	3	2	1	0
VGA						a		

	DESCRIPTION
a	This index points to one of the sequencer registers (SEQ) at I/O port address 3C5h, for the next SEQ read/write operation. These registers are described on the following pages.

SEQUENCER REGISTERS

Reset Register (SEQ00)

I/O:3C5h (R/W)
INDEX:00

	7	6	5	4	3	2	1	0
VGA							b	a

	DESCRIPTION
a	Synchronous Reset Bit 0: **0** = Follows SEQ00[1]. **1** = Allows the sequencer to run unless SEQ00[1] is zero.
b	Synchronous Reset Bit 1: **0** = Disable character clock, and display requests to the video memory and H/V sync signals. **1** = Allows the sequencer to run unless SEQ00[0] is zero.

Notes:

1. Bits 0 and 1 must both be zero (sequencer halted) before any clock select bits are changed; for example, clock selects GENMO[3:2] or SEQ01[0:3].

2. The SEQ00[0:1] bits must both be set to one for normal operation.

SEQUENCER REGISTERS

Clock Mode Register (SEQ01)

I/O:03C5h (R/W)
INDEX:01

	7	6	5	4	3	2	1	0
VGA	▨	▨	**d**	**b**			▨	**a**
					c			

DESCRIPTION	
a	8/9 Dot Clocks: **0** = Selects 9-dot character clocks. **1** = Selects 8-dot character clocks. Modes 0, 1, 2, 3, 7 use 9-dot characters. To change bit 0, GENVS[0] must be logical zero.
b	Shift 4, Shift Load bits: **00** =Loads video serializers every character clock. **01** =Loads video serializers every other character clock. **10** =Loads video serializers every fourth character clock. **11** =Loads video serializers every fourth character clock.
c	Dot Clock: **0** = Indicates dot clock is Master clock. **1** = Indicates dot clock is Master clock divided by 2. Typically, 320 and 360 horizontal modes use divide-by-2 to provide 40-column displays. To change this bit, SEQ00[0:0] must first be set to zero.
d	Screen Off: **0** = Allows CPU:CRT interleaved access to video memory. **1** = Blanks the screen and disables video-generation logic access to video memory. Allows CPU uninterrupted access to video memory.
Note: To change this register, SEQ00[1 or 0] must first be logical zero.	

	SEQUENCER REGISTERS							
	Map Mask Register (SEQ02)					I/O:3C5h (R/W) INDEX:02		
	7	6	5	4	3	2	1	0
VGA					d	c	b	a

DESCRIPTION	
a	Enable Map 0: **0** = Disables write access to memory map 0. **1** = Enables write access to memory map 0.
b	Enable Map 1: **0** = Disables write access to memory map 1. **1** = Enables write access to memory map 1.
c	Enable Map 2: **0** = Disables write access to memory map 2. **1**= Enables write access to memory map 2.
d	Enable Map 3: **0** = Disables write access to memory map 3. **1** = Enables write access to memory map 3.

Notes:

1. In 4 bit per PEL graphics modes, when the value of this register is set to '1111' (0Fh), the processor can complete a 32-bit write operation in one memory cycle.

2. In text modes, the CPU only needs to access maps 0 and 1; therefore, this register should have a value of 03h.

3. When in odd/even modes, the map mask value for maps 0 and 1 should be same as the map mask value for maps 2 and 3.

4. Memory map updating such as bit map layering can be selectively enabled or disabled using bits in this register. For pixel-oriented operations, the graphics controller provides better control.

	SEQUENCER REGISTERS							
	Character Map Select Register (SEQ03)					I/O:3C5h (R/W) INDEX:03		
	7	6	5	4	3	2	1	0
VGA			b	a	b		a	

	DESCRIPTION
a	Character Map Select B Bits 2:0.
b	Character Map Select A Bits 2:0.

Notes:

1. The above register may seem unusual in that bit 0-3 is identical to bit 4-5. This is correct and the above notation is valid.

2. Extended memory SEQ04[1] must be logical in order to enable this select function; otherwise, the first 8K of map 2 is always selected.

3. Any changes made to this register take effect at the start of the next character line on the display.

4.

*Bit Pattern	Map Selected	Offset into Map
0 0 0	0	0K
0 0 1	1	16K
0 1 0	2	32K
0 1 1	3	48K
1 0 0	4	8K
1 0 1	5	24K
1 1 0	6	40K
1 1 1	7	56K

SEQUENCER REGISTERS

	Memory Mode Register (SEQ04)					I/O:3C5h (R/W) INDEX:04		
	7	6	5	4	3	2	1	0
VGA					c	b	a	

	DESCRIPTION
a	Extended Memory Indicates 256K of video memory is present. Also enables character map selection in SEQ03.
b	Odd/Even **0** = Uses the LSB CPU address bit A0 to select which memory map to access. Even CPU addresses access maps 0 and 2; odd addresses access maps 1 and 3. **1** = Enables sequential access to video data maps for odd/even modes. Map Mask register bits SEQ02[0:3] identify which maps are to be accessed for each CPU address.
c	Chain **0** = Enables sequential data access within a bit map. Map Mask register bits SEQ02[0:3] identify which maps are to be accessed at any one time. **1** = In 256-color modes map select is by CPU address bits A0 and A1: **A1, A0 Map Selected** 0 0 0 0 1 1 1 0 2 1 1 3 When Chain is logical one, it takes priority over odd/even mode bits SEQ04[2] and GRA05[4]. Unlike odd/even mode, SEQ04[2] is the only bit used to enable chain mode (double odd/even). Chain does not affect CRTC access to video memory. Odd/even bit SEQ04[2] should be the opposite of GRA05[4].

CRT CONTROLLER REGISTERS

CRTC Index Register (CRTX)					I/O:3?4h (R/W)			
	7	6	5	4	3	2	1	0
VGA				a				

DESCRIPTION	
a	This index points to one of the internal registers of the CRT controller (CRTC) at address 3?5h, for the next CRTC read/write operation. These registers are described on the following pages.
Note: **?** = B when GENMO[0]=0 (Monochrome emulation). **?** = D when GENMO[0]=1 (Color/Graphics emulation).	

CRT CONTROLLER REGISTERS

Horizontal Total Register (CRT00)					I/O:3?5h (R/W) INDEX:00			
	7	6	5	4	3	2	1	0
VGA	a							

DESCRIPTION	
a	These bits define the active horizontal display in a scan line, including the retrace period. The value is five less than the total number of displayed characters in a scan line.
Note: **?** = B when GENMO[0]=0 (Monochrome emulation). **?** = D when GENMO[0]=1 (Color/Graphics emulation).	

	CRT CONTROLLER REGISTERS							
	Horizontal Display Enable End Register (CRT01)					I/O:3?5h (R/W) INDEX:01		
	7	6	5	4	3	2	1	0
VGA	a							

DESCRIPTION	
a	These bits define the active horizontal display in a scan line. The value is one less than the total number of displayed characters in a scan line.
Note: **?** = B when GENMO[0]=0 (Monochrome emulation). **?** = D when GENMO[0]=1 (Color/Graphics emulation).	

	CRT CONTROLLER REGISTERS							
	Start Horizontal Blanking Register (CRT02)					I/O:3?5h (R/W) INDEX:02		
	7	6	5	4	3	2	1	0
VGA	a							

DESCRIPTION	
a	These bits define the horizontal character count that represents the character count in the active display area plus the right border. In other words, the count is from the start of active display to the start of triggering the H blanking pulse.
Note: **?** = B when GENMO[0]=0 (Monochrome emulation). **?** = D when GENMO[0]=1 (Color/Graphics emulation).	

CRT CONTROLLER REGISTERS

End Horizontal Blanking Register (CRT03)					I/O:3C5h (R/W) INDEX:03			
	7	6	5	4	3	2	1	0
VGA	c	b		a				

DESCRIPTION	
a	H Blanking End bits 4-0, respectively. These are the five low-order (of six bits in total) of horizontal character count for triggering the end of the horizontal blanking pulse. The sixth bit is CRT05[7]. The character count is equal to the sum of "H blanking start" plus "H blanking pulse width".
b	Display Enable Skew: **00** =Zero-character-clock skew. **01** =One-character-clock skew. **10** =Two-character-clock skew. **11** =Three-character-clock skew.
c	Compatibility Read **0** = Enables write-only to CRT10 and CRT11. **1** = Enables read/write access to both vertical retrace start/end registers CRT10 and CRT11.
Note: **?** = B when GENMO[0]=0 (Monochrome emulation). **?** = D when GENMO[0]=1 (Color/Graphics emulation).	

	CRT CONTROLLER REGISTERS							
	Start Horizontal Retrace Register (CRT04)					I/O:3?5h (R/W) INDEX:04		
	7	6	5	4	3	2	1	0
VGA	a							

DESCRIPTION	
a	These bits define the horizontal character count at which the horizontal retrace pulse becomes active.
Note: **?** = B when GENMO[0]=0 (Monochrome emulation). **?** = D when GENMO[0]=1 (Color/Graphics emulation).	

	CRT CONTROLLER REGISTERS							
	End Horizontal Retrace Register (CRT05)					I/O:3?5h (R/W) INDEX:05		
	7	6	5	4	3	2	1	0
VGA	c	b		a				

DESCRIPTION	
a	H Retrace End bits. This is the 5-bit result from the sum of CRT04 plus the width of the horizontal retrace pulse, in the character clock units.
b	H Retrace Delay bits: These two bits skew the Horizontal Retrace pulse. **00** = Zero character clocks. **01** = One character clocks. **10** = Two character clocks. **11** = Three character clocks.
c	H Blanking End Bit 5. This is bit 5 of the 6-bit character count for the H blanking end pulse. The other five low-order bits are CRT03[4:0].
Note: **?** = B when GENMO[0]=0 (Monochrome emulation). **?** = D when GENMO[0]=1 (Color/Graphics emulation).	

CRT CONTROLLER REGISTERS								
	Vertical Total Register (CRT06)					I/O:3?5h (R/W) INDEX:06		
	7	6	5	4	3	2	1	0
VGA	a							

DESCRIPTION	
a	These are the eight low-order bits of the 10-bit vertical register. The two high-order bits are CRT07[0:5] in the CRTC overflow register. The value of this register represents the total number of H raster scans plus vertical retrace (active display, blanking), minus two scan lines.
Note: ? = B when GENMO[0]=0 (Monochrome emulation). ? = D when GENMO[0]=1 (Color/Graphics emulation).	

CRT CONTROLLER REGISTERS								
	CRTC Overflow Register (CRT07)					I/O:3?5h (R/W) INDEX:07		
	7	6	5	4	3	2	1	0
VGA	h	g	f	e	d	c	b	a

DESCRIPTION	
a	V Total Bit 8 (CRT06). Bit 8 of 10-bit vertical count for V Total (for functional description see CRT06 register).
b	End V Display Bit 8 (CRT12). Bit 8 of 10-bit vertical count for V Display enable end (for functional description see CRT12 register).
c	Start V Retrace Bit 8 (CRT10). Bit 8 of 10-bit vertical count for V Retrace start. For functional description see CRT10 register.
d	Start V Blanking Bit 8 (CRT15). Bit 8 of 10-bit vertical count for V Blanking start (for functional description see CRT15 register).

Continued on Next Page

CRT CONTROLLER REGISTERS

CRTC Overflow Register (CRT07)

I/O:3?5h (R/W)
INDEX:07

	7	6	5	4	3	2	1	0
VGA	**h**	**g**	**f**	**e**	**d**	**c**	**b**	**a**

DESCRIPTION	
e	Line Compare Bit 8 (CRT18). Bit 8 of 10-bit vertical count for Line Compare (for functional description see CRT18 register).
f	V Total Bit 9 (CRT06). Bit 9 of 10-bit vertical count for V Total (for functional description see CRT06 register).
g	End V Display Bit 9 (CRT12). Bit 9 of 10-bit vertical count for V Display enable end (for functional description see CRT12 register).
h	Start V Retrace Bit 9 (CRT10). Bit 9 of 10-bit vertical count for V Retrace start (for functional description see CRT10 register).

Notes:

1. ? = B when GENMO[0]=0 (Monochrome emulation).
   ? = D when GENMO[0]=1 (Color/Graphics emulation).
2. Various bits in this register are functionally part of other CRTC registers.

CRT CONTROLLER REGISTERS								
	Preset Row Scan Register (CRT08)					I/O:3?5h (R/W) INDEX:08		
	7	6	5	4	3	2	1	0
VGA		b		a				

DESCRIPTION	
a	Preset Row Scan bits 4:0. This register is used for software-controlled vertical scrolling in text or graphics modes. The value specifies the first line to be scanned after a V Retrace (in the next frame.) Each H Retrace pulse increments the counter by 1, up to the Maximum Scan Line value programmed by CRT09, then the counter is cleared.
b	Byte Panning Control Bits 1 and 0, respectively. Bits 6 and 5 extend the capability of byte panning (shifting) by up to three characters (for description see *H PEL Panning register ATTR13*).
Note: **?** = B when GENMO[0]=0 (Monochrome emulation) **?** = D when GENMO[0]=1 (Color/Graphics emulation)	

<table>
<tr><th></th><th colspan="8">CRT CONTROLLER REGISTERS</th></tr>
<tr><td></td><td colspan="5"><i>Maximum Scan Line Register (CRT09)</i></td><td colspan="3">I/O:3?5h (R/W)
INDEX:09</td></tr>
<tr><td></td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>VGA</td><td>d</td><td>c</td><td>b</td><td colspan="5">a</td></tr>
</table>

DESCRIPTION	
a	Maximum Scan Line bits. These bits define a value that is the actual number of scan lines per character minus one.
b	Start V Blanking bit 9 (CRT15). Bit 9 of 10-bit vertical count for V blanking start (for functional description see *CRT15 register*).
c	Line Compare bit 9 (CRT18). Bit 9 of 10-bit vertical count for line compare (for functional description see *CRT18 register*).
d	200-/400-Line Scan: **0** = Counter is incremented per scan line. **1** = Clock pulses to the row scan counter are divided by two. Effectively, this allows the lines in 200-line mode to be displayed twice before the row scan counter is incremented once. **Note:** H/V display and blanking timings etc. (in CRT00-CRT06 registers) are not affected by these bits. In ATI extended modes, scan function is configured in ATI31[5:3]
Note: **?** = B when GENMO[0]=0 (Monochrome emulation) **?** = D when GENMO[0]=1 (Color/Graphics emulation)	

CRT CONTROLLER REGISTERS

Cursor Start Register (CRT0A)

I/O:3?5h (R/W)
INDEX:0A

	7	6	5	4	3	2	1	0
VGA			**b**	**a**				

	DESCRIPTION
a	Cursor Start bits 4-0, respectively. These bits define a value that is the starting scan line (on a character row) for the line cursor. The five-bit value is equal to the actual number minus one. This value is used together with Cursor End bits CRT0B[4:0] to determine the height of the cursor. The cursor height in VGA does not wrap around (as in EGA) and is actually absent when the 'end' value is less than the 'start' value. In EGA, when the 'end' value is less, the cursor is a full block cursor, which is the same height as the character cell.
b	Cursor On/Off **0** = Cursor on. **1** = Cursor off.

Note: **?** = B when GENMO[0]=0 (Monochrome emulation).
? = D when GENMO[0]=1 (Color/Graphics emulation).

	CRT CONTROLLER REGISTERS							
	Cursor End Register (CRT0B)					I/O:3?5h (R/W) INDEX:0B		
	7	6	5	4	3	2	1	0
VGA		b		a				

DESCRIPTION	
a	Cursor End Bits 4-0, respectively. These bits define the ending scan row (on a character line) for the line cursor. In EGA, this 5-bit value is equal to the actual number of lines plus one. The cursor height in VGA does not wrap around (as in EGA) and is actually absent when the 'end' value is less than the 'start' value. In EGA when the 'end' value is less, the cursor is a full block cursor, which is the same height as the character cell.
b	Cursor Skew Bits 1 and 0, respectively. These bits define the number of characters the cursor is to be shifted to the right (skewed) from the character pointed at by the cursor location registers (CRT0E and CRT0F), in VGA mode. Skew values when in EGA mode are enclosed in brackets. **00** =Zero(zero) character skew. **01** =One (zero) character skew. **10** =Two (one) character skew. **11** =Three (two) character skew.
Note: **?** = B when GENMO[0]=0 (Monochrome emulation). **?** = D when GENMO[0]=1 (Color/Graphics emulation).	

CRT CONTROLLER REGISTERS

Start Address (High Byte) Register (CRT0C)

I/O:3?5h (R/W)
INDEX:0C

	7	6	5	4	3	2	1	0
VGA	a							

	DESCRIPTION
a	SA bits 15:8. These are the eight high-order bits of the 16-bit display buffer start location. The low-order bits are contained in CRT0D. In split screen mode, CRT0C + CRT0D points to the starting location of screen A (top half.) The starting address for screen B is always zero.

Notes:

1. ? = B when GENMO[0]=0 (Monochrome emulation).
   ? = D when GENMO[0]=1 (Color/Graphics emulation).
2. *ATI Extended modes have additional SA bits: ATI30[6], ATI23[4].

CRT CONTROLLER REGISTERS

Start Address (Low Byte) Register (CRT0D)

I/O:3?5h (R/W)
INDEX:0D

	7	6	5	4	3	2	1	0
VGA	a							

	DESCRIPTION
a	SA bits 7:0. These are the eight low-order bits of the 16-bit display buffer start location. The high-order bits are contained in CRT0C. In split screen mode, CRT0C+CRT0D points to the starting location of screen A (top half.) The starting address for screen B is always zero.

Notes:

1. ? = B when GENMO[0]=0 (Monochrome emulation).
   ? = D when GENMO[0]=1 (Color/Graphics emulation).
2. *ATI Extended modes have additional SA bits: ATI30[6], ATI23[4].

	CRT CONTROLLER REGISTERS							
	Cursor Location (High Byte) Register (CRT0E)					I/O:3?5h (R/W) INDEX:0E		
	7	6	5	4	3	2	1	0
VGA	a							

DESCRIPTION	
a	CA bits 15:8. These are the eight high-order bits of the 16-bit cursor start address. The low-order CA bits are contained in CRT0F. This address is relative to the start of physical display memory address pointed to by CRT0C+CRT0D. In other words, if CRT0C+CRT0D is changed, the cursor still points to the same character as before.

Notes:

1. ? = B when GENMO[0]=0 (Monochrome emulation)
   ? = D when GENMO[0]=1 (Color/Graphics emulation)

2. *ATI Extended modes use additional CA bits: ATI30[2], ATI23[3].

	CRT CONTROLLER REGISTERS							
	Cursor Location (Low Byte) Register (CRT0F)					I/O:3?5h (R/W) INDEX:0F		
	7	6	5	4	3	2	1	0
VGA	a							

DESCRIPTION	
a	CA bits 7:0. These are the eight low-order bits of the 16-bit cursor start address. The high-order CA bits are contained in CRT0E. This address is *relative* to the start of the physical display memory address pointed to by CRT0C+CRT0D. In other words, if CRT0C+CRT0D is changed, the cursor still points to the same character as before.

Notes:

1. ? = B when GENMO[0]=0 (Monochrome emulation).
   ? = D when GENMO[0]=1 (Color/Graphics emulation).

2. *ATI Extended modes use additional CA bits: ATI30[2], ATI23[3].

CRT CONTROLLER REGISTERS								
	Start Vertical Retrace Register (CRT10)					I/O:3?5h (R/W) INDEX:10		
	7	6	5	4	3	2	1	0
VGA	a							

DESCRIPTION	
a	Bits CRT10[7:0] are the eight low-order bits of the 10-bit vertical retrace start count. The two high-order bits are CRT07[2:7], located in the CRTC overflow register. These bits define the horizontal scan count that triggers the V retrace pulse.

Notes:

1. ? = B when GENMO[0]=0 (Monochrome emulation).
   ? = D when GENMO[0]=1 (Color/Graphics emulation).
2. This register is read/write enabled if CRT03[7] is set to one. It is write-only enabled if CRT03[7] is set to zero.

CRT CONTROLLER REGISTERS

End Vertical Retrace Register (CRT11)

I/O:3?5h (R/W)

INDEX:11

	7	6	5	4	3	2	1	0
VGA	**d**		**c**	**b**	**a**			

	DESCRIPTION
a	V Retrace End Bits 3-0. Bits CRT11[0-3] define the horizontal scan count that triggers the end of the V Retrace pulse.
b	V Retrace Interrupt Set: **0** = V Retrace interrupt cleared.
c	V Retrace Interrupt Disabled: **0** = Enable V Retrace interrupt.
d	Write Protect (CRT00-CRT07): **0** = Enables normal read/write of CRT00 to CRT07. **1** = Write protect registers CRT00 to CRT07 when in VGA mode as follows: All register bits except CRT07[4] are write protected.

Notes:

1. **?** = B when GENMO[0]=0 (Monochrome emulation)
   ? = D when GENMO[0]=1 (Color/Graphics emulation)

2. This register is read/write enabled if CRT07[7] is set to one. It is write-only enabled if CRT03[7] is set to zero.

	CRT CONTROLLER REGISTERS							
	Vertical Display Enable End Register (CRT12)					I/O:3?5h (R/W) INDEX:12		
	7	6	5	4	3	2	1	0
VGA	a							

DESCRIPTION	
a	These are the eight low-order bits of the 10-bit register containing the horizontal scan count indicating where the active display on the screen should end. The high-order bits are CRT07[1:6] in the CRT overflow register.
Note: ? = B when GENMO[0]=0 (Monochrome emulation). ? = D when GENMO[0]=1 (Color/Graphics emulation).	

	CRT CONTROLLER REGISTERS							
	Offset Register (CRT13)					I/O:3?5h (R/W) INDEX:13		
	7	6	5	4	3	2	1	0
VGA	a							

DESCRIPTION	
a	These bits define an offset value, equal to the logical line width of the screen (from the first character of the current line to the first character of the next line). Memory organization is dependent on the video mode. Bit CRT17[6] selects byte or word mode. Bit CRT14[6], which overrides the byte/word mode setting, selects Double-Word mode when it is logical one. The first character of the next line is specified by the start address (CRT0C+CRT0D) plus the offset. The offset for byte mode is 2x CRT13; for word mode, 4x; for double word mode, 8x.
Note: ? = B when GENMO[0]=0 (Monochrome emulation). ? = D when GENMO[0]=1 (Color/Graphics emulation).	

	CRT CONTROLLER REGISTERS							
	Underline Location Register (CRT14)					I/O:3?5h (R/W) INDEX:14		
	7	6	5	4	3	2	1	0
VGA		c	b	a				

DESCRIPTION	
a	H Row Scan Bits 4-0. These bits define the horizontal scan row from the top of the character line that should be used for underlining. The 5-bit value is equal to the actual number minus one.
b	Count-by-4: **0** = Character clock is used unmodified at the memory address counter, for byte addressing. **1** = Character clock is divided-by-4 at the clock input to the Memory address counter. Count-by-4 clocks are used for double-word addressing. This bit overrides divide-by-2 bit CRT17[3].
c	Double-Word Mode: **0** = Allows addressing mode to be selected by CRT17[6]. **1** = Enables double-word addressing mode. This bit overrides byte/word bit CRT17[6].
Note: **?** = B when GENMO[0]=0 (Monochrome emulation). **?** = D when GENMO[0]=1 (Color/Graphics emulation).	

CRT CONTROLLER REGISTERS

Start Vertical Blanking Register (CRT15)

I/O:3?5h (R/W)
INDEX:15

	7	6	5	4	3	2	1	0
VGA	a							

DESCRIPTION	
a	These are the eight low-order bits of the 10-bit vertical blanking start register. Bit 9 is CRT09[5]; bit 8 is CRT07[3]. The 10 bits specify the starting location of the vertical blanking pulse, in units of horizontal scan lines. The value is equal to the actual total number of displayed lines minus one.
Note: ? = B when GENMO[0]=0 (Monochrome emulation). ? = D when GENMO[0]=1 (Color/Graphics emulation).	

CRT CONTROLLER REGISTERS

End Vertical Blanking Register (CRT16)

I/O:3?5h (R/W)
INDEX:16

	7	6	5	4	3	2	1	0
VGA		a						

DESCRIPTION	
a	These bits define the point at which to trigger the end of the vertical blanking pulse. The location is specified in units of horizontal scan lines. The value to be stored in this register is the seven low-order bits of the sum of "pulse width count" plus the content of Start Vertical Blanking register (CRT15) minus one.
Note: ? = B when GENMO[0]=0 (Monochrome emulation). ? = D when GENMO[0]=1 (Color/Graphics emulation).	

CRT CONTROLLER REGISTERS

	CRT Mode Register (CRT17)				I/O:3?5h (R/W) INDEX:17			
	7	6	5	4	3	2	1	0
VGA	g	f	e	▨	d	c	b	a

DESCRIPTION	
a	Compatibility Mode: **0** = Substitutes row scan counter bit 0 as bit 13 of CRTC output during active display time. For example, this allows for compatibility with the6845controller or CGA APA modes. **1** = Enables row scan counter bit 13 as bit 13 of CRTC output.
b	Select Row Scan Counter: **0** = Selects row scan counter bit 1 (RA1) as bit 14 at the CRTC output during active display time. This substitution allows for compatibility with Hercules graphics and other 400-line graphics modes. **1** = Selects row scan counter bit 14 (RA14) as bit 14 at the CRTC output.
c	Vertical-by-2: **0** = Increments the vertical timing counter for every horizontal retrace. **1** = Increments the vertical timing counter for every two horizontal retrace pulses. It effectively doubles the vertical resolution by two, for example, to 2048 horizontal scan lines in VGA, and 1024 in EGA. **Note:** When bit 2 is logical one, other vertical register values should be adjusted as well (CRT06, CRT10, CRT12, CRT15, and CRT18).
d	Count-by-2: **0** = Increments the memory address counter for every character clock. **1** = Increments the memory address counter for every two character clocks.
e	Address Wrap: **0** = Indicates only 64K video memory is to be addressed: In word mode, address counter bits are left shifted once, so bit 13 (MA13) is wrapped around to bit 0 position at the CRTC output. **1** = Enables 256K video memory addressing: In word mode, address counter bits are rotated left by one, so bit 15 (MA15) is wrapped around to bit 0 position at the CRTC output.
f	Byte/Word Mode: **0** = Selects word mode memory addressing. The memory address is rotated left by one. **1** = Selects byte mode memory addressing.
g	H/V Retrace Enable: **0** = Disables horizontal and vertical retrace. **1** = Enables horizontal and vertical retrace.

(Continued on next page)

	CRT CONTROLLER REGISTERS							
	CRT Mode Register (CRT17)					I/O:3?5h (R/W) INDEX:17		
	7	6	5	4	3	2	1	0
VGA	g	f	e		d	c	b	a

DESCRIPTION
Notes:

1. ? = B when GENMO[0]=0 (Monochrome emulation).
   ? = D when GENMO[0]=1 (Color/Graphics emulation).
2. 640x200 mode is programmed for 100 horizontal scan lines with two row scan addresses per character row. Odd scan lines are offset in the display memory by 8K bytes.

CRT CONTROLLER REGISTERS

	Line Compare Register (CRT18)					I/O:3?5h (R/W) INDEX:18		
	7	6	5	4	3	2	1	0
VGA	a							

	DESCRIPTION
a	These bits are the eight low-order of the 10-bit line compare register. Bit 8 is CRT07[4], bit 9 is CRT09[6]. The value of this register is used to disable scrolling on a portion of the display screen, as when the split screen is active. When the vertical counter reaches this value, the memory address and row scan counters are cleared. The screen area above the line specified by this register is commonly called screen A. The screen below is screen B. Screen B cannot be scrolled, but it can be panned only together with screen A, controlled by the PEL panning compatibility bit ATTR10[5]. (For a description of this control bit see *ATTR10[5].*)
Note: **?** = B when GENMO[0]=0 (Monochrome emulation). **?** = D when GENMO[0]=1 (Color/Graphics emulation).	

GRAPHICS CONTROLLER REGISTERS

	Graphics Controller Index Register (GRAX)					I/O:3CEh (R/W)		
	7	6	5	4	3	2	1	0
VGA					a			

	DESCRIPTION
a	This index is used to address one of the internal registers of the graphics controller (GRAC) at I/O port 3CFh. These are described on the following pages.

	GRAPHICS CONTROLLER REGISTERS							
	Set/Reset Register (GRA00)					I/O:3CFh (R/W) INDEX:00		
	7	6	5	4	3	2	1	0
VGA					d	c	b	a

DESCRIPTION	
a	Set/Reset Map 0: **0** = All eight bits of buffer map 0 are to be written with zeros during CPU write if write mode is 0 (See write mode bits GRA05[1:0]), and if the enable set/reset bit GRA01[0] is a logical one. **1** = All eight bits of buffer map 1 are to be written with ones during CPU write if write mode is 0 or 3 (See write mode bits GRA05[1:0]), and if the enable set/reset bit GRA01[0] is a logical one.
b	Set/Reset Map 1: **0** = All eight bits of buffer map 1 are to be written with zeros during CPU write if write mode is 0 (See write mode bits GRA05[1:0]),and if the enable set/reset bit GRA01[1] is a logical one. **1** = All eight bits of buffer map 1 are to be written with ones during CPU write if write mode is 0 or 3 (See write mode bits GRA05[1:0]), and if the enable set/reset bit GRA01[1] is a logical one.
c	Set/Reset Map 2: **0** = All eight bits of buffer map 2 are to be written with zeros during CPU write if write mode is 0 (See write mode bits GRA05[1:0]),and if the enable set/reset bit GRA01[2] is a logical one. **1** = All eight bits of buffer map 2 are to be written with ones during CPU write if write mode is 0 or 3 (See write mode bits GRA05[1:0]), and if the enable set/reset bit GRA01[2] is a logical one.
d	Set/Reset Map 3: **0** = All eight bits of buffer map 3 are to be written with zeros during CPU write if write mode is 0 (See write mode bits GRA05[1,0]),and if the enable set/reset bit GRA01[3] is a logical one. **1** = All eight bits of buffer map 3 are to be written with ones during CPU write if write mode is 0 or 3 (See write mode bits GRA05[1:0]), and if the enable set/reset bit GRA01[3] is a logical one.

	GRAPHICS CONTROLLER REGISTERS							
	Enable Set/Reset Register (GRA01)				I/O:3CFh (R/W) INDEX:01			
	7	6	5	4	3	2	1	0
VGA					d	c	b	a

DESCRIPTION	
a	Enable Set/Reset Map 0: **0** = If write mode is 0 (GRA05[1:0]=0) CPU data is written to memory map 0. **1** = If write mode is 0 (GRA05[1:0]=0) GRA00[0] is written to all eight bits of memory map 0.
b	Enable Set/Reset Map 1: **0** = If write mode is 0 (GRA05[1:0]=0) CPU data is written to memory map 1. **1** = If write mode is 0 (GRA05[1:0]=0) GRA00[1] is written to all eight bits of memory map 1.
c	Enable Set/Reset Map 2: **0** = If write mode is 0 (GRA05[1:0]=0) CPU data is written to memory map 2. **1** = If write mode is 0 (GRA05[1:0]=0) GRA00[2] is written to all eight bits of memory map 2.
d	Enable Set/Reset Map 3: **0** = If write mode is 0 (GRA05[1:0]=0) CPU data is written to memory map 3. **1** = If write mode is 0 (GRA05[1:0]=0)GRA00[3] is written to all eight bits of memory map 3.
Note: This register has no effect on data source select when the video memory map write mode is 1, 2, or 3.	

GRAPHICS CONTROLLER REGISTERS

Color Compare Register (GRA02) — I/O:3CFh (R/W) INDEX:02

	7	6	5	4	3	2	1	0
VGA					a			

	DESCRIPTION
a	Color Compare Map bits 3-0. In Read mode (GRA05[3] being logical one), the four bits from this register are compared with the 4-bit PEL value (made up of one bit from each map), from bit positions 0 to 7. As long as the Color Don't Care bits (GRA07[0:3]) for the respective maps are logical ones, the compare takes place only on those bits of the PEL value, and the CPU reads a one for a match in that bit position. If the Color Don't Care bit for one map is logical zero, the latched data from that map is excluded from the compare, and only the remaining three bits are compared to generate the bus data.

GRAPHICS CONTROLLER REGISTERS

Data Rotate Register (GRA03)

I/O:3CFh (R/W)
INDEX:03

	7	6	5	4	3	2	1	0
VGA	▨	▨	▨	b		a		

DESCRIPTION	
a	Rotate Count Bits 2-0. Specifies the number of bit positions the CPU data is to be rotated to the right. This is done before doing the function selected by bits 3 and 4 above and subsequent bit mask select and write operations. Rotation is carried out only in write modes 0 and 3. In these two modes, the CPU data is rotated first, then operated on by the function bits GRA03[4:3], then updated by the bit mask register GRA05.
b	Function Select Bits 1 and 2: **00** =CPU data replaces latched data. **01** =CPU data ANDed with latched data. **10** =CPU data ORed with latched data. **11** =CPU data XORed with latched data. These functions are performed on the CPU data before the selected bits are updated by the bit mask register, and then written to the display buffers.

GRAPHICS CONTROLLER REGISTERS

Read Map Select Register (GRA04)

I/O:3CFh (R/W)
INDEX:04

	7	6	5	4	3	2	1	0
VGA							a	

DESCRIPTION	
a	Bits 1 and 2, respectively. Read mode 0 only: GRA controller returns the contents of one of the four latched buffer bytes to the CPU each time a CPU read loads the latches. These two bits (0 and 1) define a value that represents the bit map where the CPU is to read data — this is useful in transferring bit map data between the maps and system RAM.

Notes:

1. In Odd/Even modes, the value may be binary 00 or 01 for chained bit maps 0 and 1.
2. In mode 13h, where all maps are chained to form one map, and in read mode 1, this register is ignored.

GRAPHICS CONTROLLER REGISTERS

	Graphic Mode Register (GRA05)					I/O:3CFh (R/W) INDEX:05		
	7	6	5	4	3	2	1	0
VGA	▨	d		c	b	▨	a	

DESCRIPTION	
a	Write Mode: **00** =The CPU data byte can be written to video buffers map data latches in two dimensions: 1) Byte-oriented: to update any or all maps. 2) Pixel-oriented: to update any or all eight pixels using a predefined pixel value Updates are controlled using values in the internal registers of this graphics controller, namely GRA00-GRA08. If enable set/reset bits are all zeros, CPU data updates the latches according to the function bits GRA03[4:3], and each map is updated as masked by GRA08[7:0]. **01** =Each map is written with the contents of its respective latches. These latches are loaded by a previous CPU memory read operation. **10** =Pixel-oriented: The four low-order bits of the CPU data are combined with the pixel values from the maps according to the functions specified by GRA03[4:3], and each map is updated as masked by GRA08[7:0]. **11** =Pixel-oriented, write mode 3 involves the following data manipulations: 1) CPU data is rotated by GRA03[2:0], then logical ANDed with the Bit Mask register bits GRA08[7:0]. The result is an 8-bit mask for use in write mode 3, to determine which pixels (from step 2 below) are to be updated by the set/reset value, and which pixels are updated directly from the latches. 2) The set/reset pixel values are produced as follows: The set/reset bits GRA00[0:3] are compared with each pixel value from the latches according to function bits GRA03[4:3].
b	Read Mode: **0** =Byte-oriented: CPU reads the memory map specified by the Read Map Select register GRA04 unless SEQ04[3] is logical one(Chain). In the case where SEQ04[3] is logical one, CPU address bits A0 and A1 are used to read the specified memory map. **1** =Pixel-oriented, 4-bit value: The value is made up of one bit from each map. CPU reads the result of the comparison of this pixel value ANDed with the 4-bit color compare register value. If a bit in the Color Don't Care register (GRA07) is zero, that bit position is excluded from the compare. A match causes that position in the byte to be read out by the CPU as a one. This process is repeated for all eight pixels.

Continued on next page

	GRAPHICS CONTROLLER REGISTERS							
	Graphic Mode Register (GRA05)					I/O:3CFh (R/W) INDEX:05		
	7	6	5	4	3	2	1	0
VGA		d		c	b		a	

DESCRIPTION	
c	Odd/Even Addressing Enable Used to enable CGA emulation, this bit enables the odd/even addressing mode when it is logical one. Normally, this bit and memory mode bit SEQ04[2] are set to agree with each other in enabling odd/even mode emulation.
d	Bit 6 = 256-color Mode. Bit 5 = Shift Register Mode. These bits control how data from memory is loaded into the shift registers. M0D0:M0D7, M1D0:M1D7, M2D0:M2D7, and M3D0:M3D7 are representations of this data. The LSB bits are shifted out first: **00** = MSB LSB O/P M0D0 M0D1 M0D2 M0D3 M0D4 M0D5 M0D6 M0D7 -> C0 M1D0 M1D1 M1D2 M1D3 M1D4 M1D5 M1D6 M1D7 -> C1 M2D0 M2D1 M2D2 M2D3 M2D4 M2D5 M2D6 M2D7 -> C2 M3D0 M3D1 M3D2 M3D3 M3D4 M3D5 M3D6 M3D7 -> C3 **01** = MSB LSB O/P M1D0 M1D2 M1D4 M1D6 M0D0 M0D2 M0D4 M0D6 -> C0 M1D1 M1D3 M1D5 M1D7 M0D1 M0D3 M0D5 M0D7 -> C1 M3D0 M3D2 M3D4 M3D6 M2D0 M2D2 M2D4 M2D6 -> C2 M3D1 M3D3 M3D5 M3D7 M2D1 M2D3 M2D5 M0D7 -> C3 **10** = When GRA05[6] = 1, bit 5 is ignored — maps 0:3 data is consequently read as packed pixels. **11** = When GRA05[6] = 1, bit 5 is ignored — maps 0:3 data is consequently read as packed pixels.

	GRAPHICS CONTROLLER REGISTERS							
	Graphics Miscellaneous Register (GRA06)					I/O:3CFh (R/W) INDEX:06		
	7	6	5	4	3	2	1	0
VGA					c		b	a

DESCRIPTION	
a	Graphics/Alphanumeric Mode: **0** = Selects A/N (alphanumeric mode): display data bypasses the graphics controller and latches into the attribute controller. **1** = Selects APA (graphics) mode: color data is serialized in the shift registers before it is passed to the attribute controller.
b	Chain Odd Maps to Even: **1** = CPU address bit A0 is replaced by a higher order address bit. Even maps (0 and 2) are selected when A0 = zero; odd maps are selected when A0 = one.
c	Memory Map Read Bits 1 and 0, respectively: **00** = Maps the display buffer into processor address A0000h for 128K bytes. **01** = Maps the display buffer into processor address A0000h for 64K bytes. **10** = Maps the display buffer into processor address B0000h for 32K bytes. **11** = Maps the display buffer into processor address B8000h for 32K bytes.

GRAPHICS CONTROLLER REGISTERS

Color Don't Care Register (GRA07)

I/O:3CFh (R/W)
INDEX:07

	7	6	5	4	3	2	1	0
VGA					d	c	b	a

DESCRIPTION	
a	Ignore Map 0.
b	Ignore Map 1.
c	Ignore Map 2.
d	Ignore Map 3.

Notes:

1. A byte is latched from each memory map in a CPU read, mode 1. The color value of a pixel (PEL) is made up of a bit from each map. The 4-bit PEL value is ANDed with the four bits from this register.
2. Any bit (map x) indicated by a logical zero in this register causes the corresponding bit in the PEL value to exclude itself from the comparison with the color compare bits. The remaining bits are ANDed with the 4-bit color compare register, where a match produces a logical one for that bit position in the CPU data byte as read data.
3. For example, if register value is "1111", the entire 4-bit PEL value is compared with the color compare bits. If any bit position matches, a logical one in the corresponding bit position is generated, as the CPU reads the data.

<table>
<tr><th></th><th colspan="8">GRAPHICS CONTROLLER REGISTERS</th></tr>
<tr><td></td><td colspan="5"><i>Bit Mask Register (GRA08)</i></td><td colspan="3">I/O:3CFh (R/W)
INDEX:08</td></tr>
<tr><td></td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>VGA</td><td colspan="8">a</td></tr>
</table>

<table>
<tr><th colspan="2">DESCRIPTION</th></tr>
<tr><td>a</td><td><b>0</b> = Data is from latches: Logical zero in a bit position preserves the memory content of the four maps in the same bit position.
<b>1</b> = Data is from CPU byte: Logical one in a bit position allows updating of the four map bits that are in the same bit position. This register is used directly in write modes 0-2 only. Bit masking in write mode 3 involves the CPU data, which is described in <i>register GRA05.</i></td></tr>
</table>

<table>
<tr><th></th><th colspan="8">GRAPHICS CONTROLLER REGISTERS</th></tr>
<tr><td></td><td colspan="5"><i>Extended I/O Base Address Low (GRA50)</i></td><td colspan="3">I/O:3CFh (W)
INDEX:50</td></tr>
<tr><td></td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>ATI VGA</td><td colspan="8">a</td></tr>
</table>

<table>
<tr><th colspan="2">DESCRIPTION</th></tr>
<tr><td>a</td><td>This register contains the low eight bits (A7:A0) of the base I/O address of the ATI extended VGA registers.</td></tr>
<tr><td colspan="2"><b>Note:</b>
This register only exists on the <i>mach64</i>GX-0 and the <i>mach64</i>GX-1. For the <i>mach64</i>GX-2, <i>mach64</i>CX, and <i>mach64</i>EX chips, the ATI register address is fixed at 1CE and 1CF and cannot be programmed to other addresses.</td></tr>
</table>

GRAPHICS CONTROLLER REGISTERS

Extended I/O Base Address High (GRA51)

I/O:3CFh (W)
INDEX:51

	7	6	5	4	3	2	1	0
ATI VGA	b		▨		a			

	DESCRIPTION
a	The high four bits (A11:A8) of the base I/O address of the ATI extended VGA registers.
b	This two bit field is the index offset match field for the ATI extended VGA registers. The high two bits of the indices wrtten to the ATI extended VGA index register must match this field in order for the VGA to respond.

Note:

This register only exists on the *mach64*GX-0 and the *mach64*GX-1. For the *mach64*GX-2, *mach64*CX, and *mach64*EX chips, the ATI register address is fixed at 1CE and 1CF and cannot be programmed to other addresses.

ATTRIBUTE CONTROLLER REGISTERS

ATTR Index Register (ATTRX)

I/O:3C0h (R)
I/O:3C0h (W)

	7	6	5	4	3	2	1	0
VGA			**b**	**a**				

DESCRIPTION	
a	ATTR Index Bits 4-0. This index points to one of the internal registers of the attribute controller (ATTR) at addresses 3C1h/3C0h, for the next ATTR read/write operation. Since both the index and data registers are at the same I/O port, a pointer to the registers is necessary. This pointer can be initialized to point to the index register by a read instruction to the GENS1 register.
b	Palette Address Source: **0** = Allows the processor to load the color palette registers. **1** = Allows memory data to access the color palette registers. After loading the color palette, this bit should be set to logical one.

Notes:

1. After initialization, OUT commands toggle between writing to the ATTRX and the indexed Attribute registers.
2. The Attribute registers operate with the Palette registers to establish the video DAC PEL definition.

ATTRIBUTE CONTROLLER REGISTERS

Palette Registers 0-F (ATTR00-0F)

I/O:3C1h (R)
I/O:3C0h (W)
INDEX:0F

	7	6	5	4	3	2	1	0
VGA	▨	▨	a					

DESCRIPTION	
a	Color Index. This field maps the text attribute or graphic color input value to a display color on the screen. Color is disabled for those bits that are set to logical zero, and enabled for those bits set to logical one.

Notes:

1. The two high-order bits of a 6-bit palette register content are stored in ATTR14[3:2].
2. Color bits 4 and 5 are substituted by ATTR14[1:0] when color source select ATTR10[7] is logical one.
3. In all modes except 256-color mode, pre-mapped 4-bit pixel values are used as addresses into the 16 ATTR palette registers. These internal registers allow 16 colors to be displayed simultaneously. The actual color output is the content of these registers.
4. In 256-color mode, where 256 colors can be displayed simultaneously, these registers are used only to index into the external registers, also called the DAC color table, where the color output values are stored.
5. Modification of these 16 internal palette registers enables the user to access 64 different addresses in the DAC color table.

ATTRIBUTE CONTROLLER REGISTERS

Mode Control Register (ATTR10)

I/O:3C1h (R)
I/O:3C0h (W)
INDEX:10

	7	6	5	4	3	2	1	0
VGA	g	f	e		d	c	b	a

DESCRIPTION	
a	Graphics/*Alphanumeric Mode: **1** = Selects monochrome display. **0** = Selects A/N: alphanumeric mode. **1** = Selects APA: graphics mode.
b	Monochrome/*Color Attributes Select: **0** = Selects color display. **1** = Graphics/*Alphanumeric Mode
c	Line Graphics Enable: **0** = Sets the ninth dot to the background color: mandatory for character fonts that do not use the line graphics character codes (C0h-DFh). **1** = Enables the special line graphics character codes for monochrome emulation, and sets the ninth dot of a line graphics character to be the same as the eighth dot.
d	Blink Enable/*Background Intensity: **0** = Allows bit 7 of the character attribute to control background intensity. **1** = Allows bit 7 of the character attribute to control blinking.
e	PEL Panning Compatibility: **0** = Allows both halves of a split screen to pan together by preventing a line compare split screen function from affecting the output of PEL panning register ATTR13 and byte panning bits CRT08[6:5]. **1** = For panning only the top half of a split screen: by forcing ATTR13 output to zero until the start of the next V sync pulse when line compare condition is "true".
f	PEL Clock Select: **0** = Shift registers are clocked every dot clock. **1** = For 256-color mode 13h: eight bits of video data are packed to form a pixel.
g	Alternate Color Source: **0** = Selects palette register bits 4 and 5 (in ATTR00-0F) as source for color output bits P4 and P5. **1** = Selects ATTR14[1:0] as source for color output bits P4 and P5, respectively.

	ATTRIBUTE CONTROLLER REGISTERS							
	Overscan Color Register (ATTR11)					I/O:3C1h (R) I/O:3C0h (W) INDEX:11		
	7	6	5	4	3	2	1	0
VGA	a							

DESCRIPTION	
a	Overscan Color

Notes:

1. These bits define the color of the border (overscan) area in 80-column modes. Overscan borders are not supported in 40-column modes.
2. Refer to the description and notes for registers ATTR00-0F for information regarding how the color bits are substituted: bits 6 and 7, ATTR14[3:2], and bits 4 and 5, ATTR14[1:0].

ATTRIBUTE CONTROLLER REGISTERS

	Color Map Enable Register (ATTR12)					I/O:3C1h (R) I/O:3C0h (W) INDEX:12		
	7	6	5	4	3	2	1	0
VGA	▨	▨	b		a			

DESCRIPTION	
a	Enable Color Map bits 3-0: **0** = Disables data from maps 3-0 to be used for video output. **1** = Enables data from a specific map, maps 3-0, to be used for video output.
b	Video Status Mux bits 0-1. These are control bits for the multiplexer on color bits P0-P7. The bit selection is also indicated at GENS1[5,4] as follows: **00** =P2, P0. **01** =P5, P4. **10** =P3, P1. **11** =P7, P6.

ATTRIBUTE CONTROLLER REGISTERS

Horizontal PEL Panning Register (ATTR13)

I/O:3C1h (R)
I/O:3C0h (W)
INDEX:13

	7	6	5	4	3	2	1	0
VGA					a			

	DESCRIPTION
a	Shift Count Bits 3-0. The shift count value (0-8) indicates how many pixel positions to shift left.

COUNT VALUE	MODES 0+, 1+, 2+, 3+, 7, 7+	MODE 13	ALL OTHER MODES
0	1	0	0
1	2	-	1
2	3	1	2
3	4	-	3
4	5	2	4
5	6	-	5
6	7	3	6
7	8	-	7
8	0	-	-

Note: A/N modes 0+, 1+, 2+, 3+, and 7+ are enhanced modes with 9x16 box size resolution. A/N mode 7 has a 9x14 box size. APA mode 13 has a 320x200 screen resolution.

ATTRIBUTE CONTROLLER REGISTERS

Color Select Register (ATTR14)

I/O:3C1h (R)
I/O:3C0h (W)
INDEX:14

	7	6	5	4	3	2	1	0
VGA					**b**		**a**	

DESCRIPTION	
a	Color bits P5 and P6, respectively. These bits are the color output bits (instead of bits 5 and 4 of the internal palette registers ATTR00-0F) when alternate color source, bit ATTR10[7], is logical one.
b	Color bits P7 and P6, respectively. These two bits are the two high-order bits of the 8-bit color, used for rapid color set switching (addressing different parts of the DAC color look up table). The lower order bits are in registers ATTR00-0F.

This page intentionally left blank.

Chapter 5

VGA Register Extensions

Configuring VGA Extended Registers

The *mach64GX*, *mach64CX*, and *mach64EX* VGA controllers provide a full set of VGA extended registers for enhanced features and performance. These registers are fully described in this chapter .

The *mach64*CT does not contain the VGA extended registers. Programs which use these registers, for instance to switch pages through extended register ATI32, will not operate properly on the *mach64*CT. These programs must use the small dual page apertures on the *mach64*CT to read and write data to the screen in real mode. For a description of the small dual page apertures and how to change pages while using them, refer to the *mach64 Programmer's Guide.* Also, see CRTC_GEN_CNTL in the *mach64 Register Reference Guide* for miscellaneous bits which may be used to control the accelerator CRT controller.

For the *mach64*GX-0 and *mach64*GX-1, the I/O address and offset for ATI's VGA extended registers are user-selectable. This feature allows users to pick an unoccupied I/O address for the extended registers when there is an I/O port conflict with other peripheral hardware. On system reset, the BIOS normally writes the values for *I/O Address* (A11:A0) and *Offset* (O1:O0) to locations 3CEh offset 50h and 51h. These registers are write-only.

The default values for address and offset are "1CE" and "2" respectively, as illustrated below:

Address: 3CEh **Offset:** 50h

Bit 7 **Bit 0**

A7	A6	A5	A4	A3	A2	A1	A0

Address: 3CEh **Offset:** 51h

Bit 7 **Bit 0**

O1	O0	“0”	“0”	A11	A10	A9	A8

The following code will program “1CE” as the I/O address, and “2” as the index offset:

```
MOV  DX, 3CE
MOV  AX, CE50          ; CE = address bits A7 to A0.
OUT  DX, AX
MOV  AX, 8151          ; 81 = offset bits 1, 0 and address bits
                       ; A11 to A8.
OUT  DX, AX
```

For the *mach64*GX-2, *mach64*CX, and *mach64*EX chips, the ATI register address is fixed at 1CE and 1CF and cannot be programmed to other addresses.

VGA Extended Registers — By Name

The following table lists all ATI’s VGA extended registers and the page numbers of their descriptions.

Name	Type	Description	Page
ATIX	R/W	ATI Index	*5-4*
ATI01	R/W	ATI Register 1	-
ATI02	R/W	ATI Register 2	-
ATI03	R/W	ATI Register 3	-
ATI04	R/W	ATI Register 4	-
ATI05	R/W	ATI Register 5	*5-5*
ATI06	R/W	ATI Register 6	*5-6*
ATI20	R/W	ATI Register 20	*5-7*
ATI23	R/W	ATI Register 23	*5-8*
ATI24	R/W	ATI Register 24	*5-9*
ATI25	R/W	ATI Register 25	*5-10*
ATI26	R/W	ATI Register 26	*5-11*
ATI28	R	ATI Register 28	*5-12*
ATI29	R	ATI Register 29	*5-13*
ATI2B	R/W	ATI Register 2B	*5-14*
ATI2D	R/W	ATI Register 2D	-
ATI2E	R/W	ATI Register 2E	-
ATI30	R/W	ATI Register 30	*5-15*
ATI31	R/W	ATI Register 31	*5-16*
ATI32	R/W	ATI Register 32	*5-17*
ATI33	R/W	ATI Register 33	*5-18*

Name	Type	Description	Page
ATI34	R/W	ATI Register 34	*5-19*
ATI35	R/W	ATI Register 35	*5-20*
ATI36	R/W	ATI Register 36	*5-21*
ATI37	R	ATI Register 37	-
ATI38	R/W	ATI Register 38	*5-22*
ATI39	R/W	ATI Register 39	*5-23*
ATI3A	R/W	ATI Register 3A	-
ATI3B	R/W	ATI Register 3B	*5-24*
ATI3C	R	ATI Register 3C	*5-25*
ATI3D	R/W	ATI Register 3D	*5-26*
ATI3E	R/W	ATI Register 3E	*5-27*
ATI3F	R/W	ATI Register 3F	*5-28*

	ATI EXTENDED REGISTERS							
	ATI Index Register (ATIX)					I/O:* (R/W)		INDEX:
	7	6	5	4	3	2	1	0
ATI VGA	h	g	f	e	d	c	b	a

DESCRIPTION	
a	Index Bit I0
b	Index Bit I1
c	Index Bit I2
d	Index Bit I3
e	Index Bit I4
f	Index Bit I5
g	Offset Bit O0
h	Offset Bit O1

*The I/O address and offset is programmed as illustrated on page 5-1.
ATI Extended registers must have both I/O address and offset configured before being accessed.

Note:

- Assuming that the I/O address for the extended registers has been configured as illustrated in the example at the beginning of this chapter, (i.e., an address of 1CEh with an index offset of 2h), the steps for writing a value of "FF" to the extended register at index "3E" (ATI3E) are as follows:

(*a*) Obtain the offset for ATI3E by concatenating the index offset "2" (bits O1-O0) to the register index "3E" (bits I5-I0). For this example, it is "1011 1110" (BE):

Address: 1CEh **Offset:** 2h

Bit 7 **Bit 0**

O1	O0	I5	I4	I3	I2	I1	I0

(*b*) Write "FF" to the extended register ATI3E by specifying "FFBE" in the program code as follows:

```
MOV  DX, 1CE
MOV  AX, FFBE
OUT  DX, AX
```

	ATI EXTENDED REGISTERS							
	ATI Register 5 (ATI05)					I/O:* (R/W)		INDEX:
	7	6	5	4	3	2	1	0
ATI VGA	b	a						

DESCRIPTION	
a	Delay latch memory read data in VGA planar mode by half the period of the memory clock.
b	Cursor blink rate select: **0** = Normal blink rate (VGA standard) **1** = Half normal blink rate.

Notes:

1. *The port address of ATI Extended registers is user-programmable. Refer to *Note* on page 5-4 for details.
2. ATI05[3:0] can be programmed only when ATI2E[4] is logical zero.

	ATI EXTENDED REGISTERS							
	ATI Register 6 (ATI06)					I/O:* (R/W)		INDEX:
	7	6	5	4	3	2	1	0
ATI VGA						a		

DESCRIPTION	
a	Text mode character FIFO depth. Power-up default is 2.

Note:

*The port address of ATI Extended registers is user-programmable. Refer to *Note* on page 5-4 for details.

	ATI EXTENDED REGISTERS							
	ATI Register 20 (ATI20)					I/O:* (R/W)		INDEX:
	7	6	5	4	3	2	1	0
ATI VGA		c		b	a			

	DESCRIPTION
a	Display FIFO Bits 3:0. These bits select the video FIFO depth at which Display Request changes from low priority to high priority in the memory controller. Power-up default is 8.
b	16-bit ROM Access: 1 = Enables 16-bit ROM access.
c	DAC extended address select bits RS3:2.

Note:

*The port address of ATI Extended registers is user-programmable. Refer to *Note* on page 5-4 for details.

	ATI EXTENDED REGISTERS							
	ATI Register 23 (ATI23)					I/O:* (R/W)		INDEX:
	7	6	5	4	3	2	1	0
ATI VGA				b		a		

DESCRIPTION	
a	**GX-0/GX-1:** 16-Bit ROM Access Bits 2:0. ROM Access Time bit (in single ROM, 16-bit mode). **CX/EX/GX-2:** Horizontal sync skew: 0 = No skew 1 = Skew by 1 character clock 2 = Skew by 2 character clocks 3 = Skew by 3 character clocks 4-7 = *Reserved*
b	ATI-Ext CRTC SA Bit 17. This is bit 17 of the display buffer start address (SA) when in ATI extended modes. See CRT0E and CRT0F for descriptions.

Note:

*The port address of ATI Extended registers is user-programmable. Refer to *Note* on page 5-4 for details.

	ATI EXTENDED REGISTERS							
	ATI Register 24 (ATI24)					I/O:* (R/W)		INDEX:
	7	6	5	4	3	2	1	0
ATI VGA	h	g	f	e	d	c	b	a

DESCRIPTION	
a	ROM Page 0 Bit 0.
b	ROM Page 0 Bit 1.
c	ROM Page 0 Bit 2.
d	ROM Page 0 Bit 3.
e	ROM Page 1 Bit 0.
f	ROM Page 1 Bit 1.
g	ROM Page 1 Bit 2.
h	ROM Page 1 Bit 3.

Note:

*The port address of ATI Extended registers is user-programmable. Refer to *Note* on page 5-4 for details.

	ATI EXTENDED REGISTERS							
	ATI Register 25 (ATI25)					I/O:* (R/W)		INDEX:
	7	6	5	4	3	2	1	0
ATI VGA	h	g	f	e	d	c	b	a

DESCRIPTION	
a	ROM Page 2 Bit 0.
b	ROM Page 2 Bit 1.
c	ROM Page 2 Bit 2.
d	ROM Page 2 Bit 3.
e	ROM Page 3 Bit 0.
f	ROM Page 3 Bit 1.
g	ROM Page 3 Bit 2.
h	ROM Page 3 Bit 3.

Note:

*The port address of ATI Extended registers is user-programmable. Refer to *Note* on page 5-4 for details.

	ATI EXTENDED REGISTERS							
	ATI Register 26 (ATI26)					I/O:* (R/W)		INDEX:
	7	6	5	4	3	2	1	0
ATI VGA	b	a						

DESCRIPTION	
a	Solid Underline: **0** = Dashed underline in monochrome text mode. **1** = Solid Underline in monochrome text mode.
b	Forced Read 3CCh: **0** = Normal read back operation. **1** = Force data bits GENMO[7:1] to logical 0 during read operation of 3CCh.

Note:

*The port address of ATI Extended registers is user-programmable. Refer to *Note* on page 5-4 for details.

	ATI EXTENDED REGISTERS							
	ATI Register 28 (ATI28)					I/O:* (R)		INDEX:
	7	6	5	4	3	2	1	0
ATI VGA							b	a

DESCRIPTION	
a	Vertical Line Counter Bit 8.
b	Vertical Line Counter Bit 9.

Note:

*The port address of ATI Extended registers is user-programmable. Refer to *Note* on page 5-4 for details.

	ATI EXTENDED REGISTERS							
	ATI Register 29 (ATI29)					I/O:* (R)		INDEX:
	7	6	5	4	3	2	1	0
ATI VGA	a							

DESCRIPTION	
a	Vertical Line Counter Bits 7:0.

Note:

*The port address of ATI Extended registers is user-programmable. Refer to *Note* on page 5-4 for details.

	ATI EXTENDED REGISTERS							
	ATI Register 2B (ATI2B)					I/O:* (R/W)		INDEX:
	7	6	5	4	3	2	1	0
ATI VGA				e	d	c	b	a

DESCRIPTION	
a	Video Zero Wait-State Enable: **1** = Enable zero wait-state support for video memory write.
b	BIOS Zero Wait-State Enable: **1** = Enable zero wait-state support for BIOS read.
c	I/O Zero Wait State Enable: **1** = Enable I/O zero wait state.
d	Double scan lock enable: **1** = Lock the CRTC9[7] bit from being altered.
e	Lock DAC Write: **1** = Lock DAC write select signal.

Note:

*The port address of ATI Extended registers is user-programmable. Refer to *Note* on page 5-4 for details.

	ATI EXTENDED REGISTERS							
	ATI Register 30 (ATI30)				I/O:* (R/W)		INDEX:	
	7	6	5	4	3	2	1	0
ATI VGA		d	c	b	a			

DESCRIPTION	
a	1M Memory Support: **1** = Indicates 1M video RAM is reserved for VGA.
b	Video Memory Size: **0** = 256K video RAM is reserved for VGA. **1** = 512K video RAM is reserved for VGA.
c	ATI-Ext 256-Color Mode Select: **1** = Enables 256-color mode in ATI extended mode.
d	ATI-Ext CRTC SA Bit 16: This is bit 16 of the display buffer start address (SA) when in ATI extended modes. See CRT0C and CRT0D for descriptions.

Note:

*The port address of ATI Extended registers is user-programmable. Refer to *Note* on page 5-4 for details.

ATI EXTENDED REGISTERS

	ATI Register 31 (ATI31)					I/O:* (R/W)		INDEX:
	7	6	5	4	3	2	1	0
ATI VGA		b	a					

	DESCRIPTION
a	Scan Function Bits 2:0. **000** = Normal scanning. **001** = APA mode: Enables double scanning in lieu of CRT09[7]. **010** = APA mode: Enables 3 of 4 scanning. **101** = A/N mode: Enables double scanning in lieu of CRT09[7]. **110** = A/N mode: Enables 3 of 4 scanning.
b	V Timings Divide-by-2: **0** = Vertical timings at current clock rate. **1** = Divides vertical timing parameters by 2.

Note:

*The port address of ATI Extended registers is user-programmable. Refer to *Note* on page 5-4 for details.

	ATI EXTENDED REGISTERS							
	ATI Register 32 (ATI32)					I/O:* (R/W)		INDEX:
	7	6	5	4	3	2	1	0
ATI VGA	c			b				a

DESCRIPTION	
a	CPU Address Read Paging bit — used with bits 7:5, must be set to logical zero.
b	**0000** = 1st 64K block or 1st 128K block. **0001** = 2nd 64K block or 2nd 128K block. **0010** = 3rd 64K block or 3rd 128K block. **0011** = 4th 64K block or 4th 128K block. **0100** = 5th 64K block. **0101** = 6th 64K block. **0110** = 7th 64K block. **0111** = 8th 64K block. **1000** = 9th 64K block. CPU Address Write Paging bits 3:0. If ATI3E[3] is logical zero, bits ATI32[3:1] are used to control both CPU read and write paging.
c	**000** = 1st 64K block or 1st 128K block. **001** = 2nd 64K block or 2nd 128K block. **010** = 3rd 64K block or 3rd 128K block. **011** = 4th 64K block or 4th 128K block. **100** = 5th 64K block. **101** = 6th 64K block. **110** = 7th 64K block. **111** = 8th 64K block. CPU Address Read Paging bits 2:0. If ATI3E[3] is logical one, bits ATI32[5:7] are used to control CPU read paging; while ATI32[1:3] are used to control CPU write paging. Read/write bit. Any status can be stored here and read back later.

Notes:

1. *The port address of ATI Extended registers is user-programmable. Refer to *Note* on page 5-4 for details.

2. CPU page addressing control is affected by ATI3D[2], which is for enabling 128K CPU addressing.

	ATI EXTENDED REGISTERS							
	ATI Register 33 (ATI33)					I/O:* (R/W)		INDEX:
	7	6	5	4	3	2	1	0
ATI VGA	b	a						

DESCRIPTION	
a	4-bit PEL, 1 bit/map: **1** = Enables map memory mapping for 16-color 1024x768 mode 55h.
b	Double Scan Enable: **0** = Scans at the selected clock rate. **1** = Enables double scanning when the screen size is programmed to be between 150 lines.

Note:

*The port address of ATI Extended registers is user-programmable. Refer to *Note* on page 5-4 for details.

	ATI EXTENDED REGISTERS							
	ATI Register 34 (ATI34)					I/O:* (R/W)		INDEX:
	7	6	5	4	3	2	1	0
ATI VGA	g	f	e	d	c	b		a

DESCRIPTION	
a	Enable CGA Emulation **1** = Enables CGA emulation.
b	Write Protect CRT09[0:4, 7] **1** = Write-protects registers CRT09[0:4, 7].
c	Write-Protect V Timing Registers **1** = Write-protects all vertical timing related registers: CRT06 CRT07[0-3,5-7] CRT09[5] CRT10, CRT11[0-3] CRT12 CRT15 CRT16.
d	Write-Protect CRT0A-CRT0B **1** = Write protects the cursor start/end registers CRT0A/CRT0B.
e	Write-Protect CRT00-CRT07 **1** = Replaces CRT11[7] as the write-protect bit for registers CRT00-CRT07 (except bit 4 of CRT07, which is not protected).
f	Write-Protect CRT08[0-6], CRT14[0-4] **1** = Write-protects registers CRT08[0:6] and CRT14[0:4].
g	CRT11[7] Override **1** = Write-protect bit CRT11[7] is ignored. See *CRT11[7]* for a list of registers to protect from a write operation.

Note:

*The port address of ATI Extended registers is user-programmable. Refer to *Note* on page 5-4 for details.

	ATI EXTENDED REGISTERS							
	ATI Register 35 (ATI35)					I/O:* (R/W)		INDEX:
	7	6	5	4	3	2	1	0
ATI VGA	f	e	d	c	▨	▨	b	a

DESCRIPTION	
a	Blanking Signal Select: **1** = Selects the Display Enable signal as the output blanking signal.
b	Blanking Polarity Invert: **1** = Inverts polarity of blanking signal BLANKB.
c	Anti-alias Fonts Enable: **1** = Enables eight simultaneous fonts.
d	Cursor Blinking Disable: **1** = Disables cursor blinking
e	CGA Cursor Start/End Address: **1** = Adds a value of `5' to both cursor start/end registers, for CGA emulation.
f	VGA Overscan Output Enable: **1** = Used as an overlay input signal to the DAC to generate an overscan feature that is independent of the DAC palette values. (ATI68800-6).

Note:

*The port address of ATI Extended registers is user-programmable. Refer to *Note* on page 5-4 for details.

	ATI EXTENDED REGISTERS							
	ATI Register 36 (ATI36)					I/O:* (R/W)		INDEX:
	7	6	5	4	3	2	1	0
ATI VGA	e	d	c			b		a

	DESCRIPTION
a	CRTC Display Address Counter Enable: **1** = Enables VGA CRTC display address counter above 64K address space.
b	Linear Addressing, 256-color Mode: **1** = Selects linear addressing, 256-color mode.
c	Vertical Interrupt Enable: **1** = Enables vertical interrupt.
d	Linear Addressing, Text Mode: **1** = Selects linear addressing, text mode.
e	Screen Blanking Disable (CGA/Hercules Modes): **1** = Disables function of bit 3 in the CGA and in the Hercules mode control registers GENMC (at I/O ports 3D8h & 3B8h respectively, to prevent screen blanking when in these modes).

Note:

*The port address of ATI Extended registers is user-programmable. Refer to *Note* on page 5-4 for details.

	ATI EXTENDED REGISTERS							
	ATI Register 38 (ATI38)					I/O:* (R/W)		INDEX:
	7	6	5	4	3	2	1	0
ATI VGA	f	e			d	c	b	a

DESCRIPTION	
a	Write-Protect ATTR00 - 0F. **1** = Disable I/O writes to ATTR00 - 0F palette registers.
b	Write-Protect ATTR11. **1** = Disable I/O writes to ATTR11 overscan register.
c	Write-protect VGA registers. **1** = Disable I/O writes to all VGA registers except: CRTC0C, CRTC0D (Start Address registers). CRTC0A (Cursor Start register) CRTC0B (Cursor End register).
d	Write Protect Register at I/O port 3C2h. **1** = Disable I/O writes to 3C2h register.
e	**0** = Input video clock undivided. **1** = input video clock divided by 2.
f	General Purpose Read/Write bit.

Note:

*The port address of ATI Extended registers is user-programmable. Refer to *Note* on page 5-4 for details.

ATI EXTENDED REGISTERS

	ATI Register 39 (ATI39)					I/O:* (R/W)		INDEX:
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ATI VGA	**d**		**c**		**b**		**a**	

DESCRIPTION	
a	Clock Frequency Synthesizer select bit 2 *(See ATI3E).*
b	ROM Address Decode Bits 1 and 0, respectively: **00** = Enables 32K BIOS size, starting at C0000h. **01** = Enables 28K BIOS size, starting at C0000h. **10** = Enables 24K BIOS size, starting at C0000h. **11** = Enables 24K BIOS size, starting at C0000h.
c	General Purpose Read/Write bit.
d	Write Protects CRT18 (Line Compare): **1** = Disables write to CRT18 (Line Compare register).

Note:

*The port address of ATI Extended registers is user-programmable. Refer to *Note* on page 5-4 for details.

	ATI EXTENDED REGISTERS							
	ATI Register 3B (ATI3B)					I/O:* (R/W)		INDEX:
	7	6	5	4	3	2	1	0
ATI VGA	a							

DESCRIPTION	
a	General Purpose Read/Write bits 7 to 0.

Note:

*The port address of ATI Extended registers is user-programmable. Refer to *Note* on page 5-4 for details.

	ATI EXTENDED REGISTERS							
	ATI Register 3C (ATI3C)					I/O:* (R)		INDEX:
	7	6	5	4	3	2	1	0
ATI VGA	a							

DESCRIPTION	
a	ASIC test control bits.

Note:

*The port address of ATI Extended registers is user-programmable. Refer to *Note* on page 5-4 for details.

	ATI EXTENDED REGISTERS							
	ATI Register 3D (ATI3D)					I/O:* (R/W)		INDEX:
	7	6	5	4	3	2	1	0
ATI VGA					c	b		a

DESCRIPTION	
a	Composite Sync Polarity Select: **1** = Selects Composite Sync Polarity output
b	128K CPU Address: **1** = Enables CPU addressing of 128K from A0000h to BFFFFh; all CPU Adrs Read page control bits in ATI32 have 128K per block operations.
c	Composite Sync Select: **1** = Selects Composite Sync output instead of Horizontal Sync.

Note:

*The port address of ATI Extended registers is user-programmable. Refer to *Note* on page 5-4 for details.

	ATI EXTENDED REGISTERS							
	ATI Register 3E (ATI3E)					I/O:* (R/W)		INDEX:
	7	6	5	4	3	2	1	0
ATI VGA				d	c		b	a

DESCRIPTION	
a	R/W V Display End Register: **1** = Allows programming of register CRT12 (Vertical Display End); even in Double Scan mode.
b	Interlace Operation: **1** = Enables interlace operation.
c	Read/Write Paging Select: **0** = Selects CPU read/write paging, as defined in ATI32[1:3]. **1** = Selects CPU read/write paging, as defined in ATI32[1:3, 5:7].
d	Clock Frequency Synthesizer Select bit 3 (See ATI39).

Note:

*The port address of ATI Extended registers is user-programmable. Refer to *Note* on page 5-4 for details.

	ATI EXTENDED REGISTERS							
	ATI Register 3F (ATI3F)					I/O:* (R/W)		INDEX:
	7	6	5	4	3	2	1	0
ATI VGA					b			a

DESCRIPTION	
a	1 = Disables zero wait state in planar mode.
b	1 = Selects the number of MCLK delay in 16-bit Planar mode memory operation before latching the first CPU data.

Note:

*The port address of ATI Extended registers is user-programmable. Refer to *Note* on page 5-4 for details.

Appendix A

BIOS Function Calls

VGA Controller

AH = 0 ; **set video mode** (AL = video mode)

AL	MODE/TYPE	RESOLUTION	DIM/COLOR	START ADDRESS
	IBM Compatible Modes:			
00h	color/alpha	640x200	40x25/BW	B800h:0
01h	color/alpha	640x200	40x25/16	B800h:0
02h	color/alpha	640x200	80x25/BW	B800h:0
03h	color/alpha	640x200	80x25/16	B800h:0
04h	color/graphics	320x200	40x25/4	B800h:0
05h	color/graphics	320x200	40x25/BW	B800h:0
06h	color/graphics	320x200	80x25/BW	B800h:0
07h	mono/alpha	720x350	80x25/BW	B000h:0
0Dh	color/graphics	320x200	40x25/16	A000h:0
0Eh	color/graphics	640x200	80x25/16	A000h:0
0Fh	mono/graphics	640x350	80x25/BW	A000h:0
10h	color/graphics	640x350	80x25/16	A000h:0
11h	color/graphics	640x480	80x30/BW	A000h:0
12h	color/graphics	640x480	80x30/16	A000h:0
13h	color/graphics	320x200	80x25/256	A000h:0

AL	MODE/TYPE	RESOLUTION	DIM/COLOR	START ADDRESS
	ATI Enhanced Modes:			
21h	color/alpha	800x400	100x25	B800h:0
22h	color/alpha	800x480	100x30	B800h:0
23h	color/alpha	1056x200	132x25/16	B800h:0
33h	color/alpha	1056x352	132x44/16	B800h:0
55h	color/graphics	1024x768	128x48/16	A000h:0
61h	color/graphics	640x400	80x25/256	A000h:0
62h	color/graphics	640x480	80x30/256	A000h:0
63h	color/graphics	800x600	100x42/256	A000h:0
64h	color/graphics	1024x768	128x48/256	A000h:0
6Ah	color/graphics	800x600	100x42/16	A000h:0

AH = 1 **; set cursor type**

CH = start line of cursor
CL = end line of cursor
CX = 1F00h to turn off cursor

AH = 2 **; set current cursor position**

BH = page number of the desired page
DH, DL = row and column of cursor

AH = 3 **; read current cursor position at the specified page**

BH = page number of the desired page
on exit:
CH, CL = cursor type
DH, DL = row, column of cursor at the specified page

AH = 4 ; **read current light pen position** (VGA does not support light pen)

AH = 5 ; **select active display page**

AL = page number to be active

AH = 6 ; **scroll active page up**

AL = number of lines to be scrolled
= 0 ; blanks the whole window
BH = attribute of blanked line
CH, CL = row, column of upper left hand corner of scrolling window
DH, DL = row, column of lower right hand corner of scrolling window

AH = 7 ; **scroll active page down**

AL = number of lines to be scrolled
= 0 ; blanks the whole window
BH = attribute of blanked line
CH, CL = row, column of upper left hand corner of scrolling window
DH, DL = row, column of lower right hand corner of scrolling window

AH = 8 ; **read character/attribute at current active cursor position**

BH = page number of the desired page
on exit:
AL = character
AH = attribute (for text mode only)

AH = 9 ; **write character/attribute at current cursor position of a specified page**

AL = character to be written
BL = attribute of character
BH = page number
CX = count of character to write

AH = 0Ah ; **write character at current cursor position of a specified page**

AL = character to be written
BH = page number
CX = count of character to write

AH = 0Bh ; **set color palette, valid for modes 4 and 5 only**

BH = 0 ; selects the background color
BL = color value used with that color id
= 1 ; selects the palette to be used
BL = 0 ; palette value is GREEN(1)/RED(2)/BROWN(3)
= 1 ; palette value is CYAN(1)/MAGENTA(2)/WHITE(3)

AH = 0Ch ; **write dot (graphics mode)**

BH = page number
DX, CX = row, column of dot position

AL = color value of dot (if bit 7 of AL is ON, the color value will be XOR'd with the current
value of the dot)

AH = 0Dh **; read dot (graphics mode)**

BH = page number
DX, CX = row, column of dot position
on exit
AL = color value of dot

AH = 0Eh **; write teletype to active page**

AL = character to write
BL = foreground color in graphics mode

AH = 0Fh **; return current video setting**

on exit:
AL = current mode
AH = number of column (in characters) on screen
BH = current active display page

AH = 10h **; set palette registers**

AL = 0 ; set individual palette register
BL = palette register
BH = palette value
AL = 1 ; set overscan register
BH = palette value
AL = 2 ; set all palette and overscan registers
ES:DX = pointer to palette value table (17 bytes long), bytes 0 - 15 are
palette
values for 16 palette registers, byte 16 is palette value for the
overscan register
AL = 3 ; toggle between intensity/blinking bit
BL = 0 ; set intensity on
BL = 1 ; set blinking on
AL = 7 ; read individual palette register
BL = palette register
on exit:
BH = palette value
AL = 8 ; read overscan register
on exit:
BH = overscan value
AL = 9 ; read all palette and overscan registers
ES:DX = pointer to 17-byte buffer
on exit:
ES:DX = pointer to palette value table (17 bytes long), bytes 0 - 15 are
palette
values for 16 palette registers, byte 16 is palette value for the

```
            overscan
                               register
            AL = 10h   ; set a color register
                       BX  =  color register
                       DH  =  red value
                       CH  =  green value
                       CL  =  blue value
            AL = 12h   ; set a block of color registers
                       BX  =  first color register to be set
                       CX  =  total number of color registers to be set
                       ES:DX =  pointer to table of color register values in red, green, blue, red,
            green,
                               blue ,... format
            AL = 13h   ; set color pages (only valid for 16 color modes)
                       BL  = 0    ; select color page mode
                       BH  = 0    ; select 4 pages of 64 color registers each
                       BH  = 1    ; select 16 pages of 16 color registers each
                       BL  = 1    ; select color page
                       BH  =  color page number
            AL = 15h   ; read a color register
                       BX  =  color register
                       on exit:
                       DH  =  red value
                       CH  =  green value
                       CL  =  blue value
            AL =17h    ; read a block of color registers
                       BX  =  first color register to be set
                       CX  =  total number of color registers to be set
                       ES:DX =  pointer to buffer to store the color register values
                       on exit:
                       ES:DX =  pointer to table of color register values in red, green, blue, red,
            green,
                               blue, ..., format
            AL = 1Ah   ; read current color page information
                       BL  =  current color page mode
                       BH  =  current color page
            AL = 1Bh   ; change color values to gray shades
                       BX  =  first color register to be changed
                       CX  =  total number of color registers to be changed

AH=11h      ; character generator routines

            AL  =  00  ; load user specified character set
                       ES:BP =  pointer to character table
                       CX  =  number of characters to be stored
                       DX  =  character of offset into current table
                       BL  =  block to load
                       BH  =  bytes per character
```

```
AL  =  01   ; load 8x14 character set
            BL  =  block to load
AL  =  02   ; load 8x8 character set
            BL  =  block to load
AL  =  03   ; set block specifier
            BL  =  character generator block specifier
AL  =  04   ; load 8x16 character set
            BL  =  block to load
```

The function AL = 1? is similar in function to AL = 0? except the number of rows on the screen is recalculated.

```
AL  =  10h  ; load user specified character set
            ES:BP =  pointer to character table
            CX  =  number of characters to be stored
            DX  =  character of offset into current table
            BL  =  block to load
            BH  =  bytes per character
AL  =  11h  ; load 8x14 character set
            BL  =  block to load
AL = 12h    ; load 8x8 character set
            BL  =  block to load
AL = 14h    ; load 8x16 character set
            BL  =  block to load
AL = 20h    ; update alternative character generator pointer (INT 1F)
            ES:BP  =  pointer to table
AL = 21h    ; update alternative character generator pointer (INT 43)
            ES:BP  =  pointer to table
            CX  =  bytes per character
            BL  =  row specifier
                =  0  ; DL = rows
                =  1  ; rows = 14
                =  2  ; rows = 25
                =  3  ; rows = 43
AL = 22h    ; update alternative character generator pointer (INT 43) with the 8x14 char-
acter
            ; generator in ROM
AL = 23h    ; update alternative character generator pointer (INT 43) with the 8x8 charac-
ter
            ; generator in ROM
AL = 24h    ; update alternative character generator pointer (INT 43) with the 8x16 char-
acter
            ; generator in ROM
AL = 30h    ; return EGA character generator information
            BH  =  0  ; return current INT 1F pointer
                =  1  ; return current INT 43 pointer
                =  2  ; return pointer to 8x14 character generator
                =  3  ; return pointer to 8x8 character generator (lower)
                =  4  ; return pointer to 8x8 character generator (upper)
```

```
                    = 5   ; return pointer to alternate 9x14 alpha
                    = 6   ; return pointer to 8x16 character generator
                    = 7   ; return pointer to alternate 9x16 alpha
                 on exit:
                 ES:BP = pointer to table as requested
                 CX  =  points (pixel column per char)
                 DL  =  rows (scan line per char)
```

AH = 12h ; return current EGA settings/print screen routine selection

```
        BL = 10h  ; return EGA information
                  on exit:
                  BH  =  0  ; color mode in effect
                      =  1  ; monochrome mode in effect
                  BL  =  3  ; 256k video memory installed (always return 3)
                  CH  =  simulated value of feature bits
                  CL  =  simulated EGA/VGA dip switch setting
        BL = 20h  ; select alternate print screen routine for EGA graphics mode
        BL = 30h  ; select number of scan lines for alpha modes
                  AL  =  0  ; 200 scan lines
                      =  1  ; 350 scan lines
                      =  2  ; 400 scan lines
                  on exit:
                  AL = 12h  ; function supported
        BL = 31h  ; default palette loading during mode set
                  AH  =  0
                  AL  =  0  ; enable
                      =  1  ; disable
                  on exit:
                  AL = 12h  ; function supported
        BL = 32h  ; video controller
                  AL  =  0  ; enable video controller
                      =  1  ; disable video controller
                  on exit:
                  AL = 12h  ; function supported
        BL = 33h  ; summing of color registers to gray shades
                  AL  =  0  ; enable summing
                      =  1  ; disable summing
                  on exit:
                  AL = 12h  ; function supported
        BL = 34h  ; cursor emulation
                  AL  =  0  ; enable cursor emulation
                      =  1  ; disable cursor emulation
                  on exit:
                  AL = 12h  ; function supported
        BL = 36h  ; video screen on/off
                  AL  =  0  ; video screen on
                      =  1  ; video screen off
                  on exit:
```

```
                AL = 12h  ; function supported
BX=5506h  ; VGAWONDER BIOS extension
                AL  =  video mode
                BP   = 0FFFFh
                DI  =  0
                SI  =  0
                on exit:
                if BP is not equal to 0FFFFh
                      then ES:BP = pointer to parameter table
                if SI is not equal to 0
                      then ES:SI = pointer to parameter table supplement
```

AH = 13h ; write string to specified page

```
ES:BP = pointer to string
CX  =  length of string
BH  =  page number
DH,DL =   starting row and column of cursor in which the string is placed
AL  =  0   ; cursor is not moved
           BL  =  attribute
           string = (char, char, char, char, ...)
AL  =  1   ; cursor is moved
           BL  =  attribute
           string = (char, char, char, char, ...)
AL  =  2   ; cursor is not moved
           string = (char, attr, char, attr, ...)
AL  =  3   ; cursor is moved
           string = (char, attr, char, attr, ...)
```

AH=1Ah ; read display combination code

```
AL  =  0   ; read current display combination information
           on exit
           AL  =  1Ah
           BL  =  current active display code
           BH  =  alternate display code
           Display codes
           00 - No display
           01 - MDA mode
           02 - CGA mode
           04 - EGA in color mode
           05 - EGA in monochrome mode
           07 - VGA with analog monochrome monitor
           08 - VGA with analog color monitor
```

AH=1Bh ; return VGA functionality and state information

```
BX  =  0   ;
           ES:DI =   pointer to buffer used to store the functionality and state informa-
tion
```

(mininum 64 bytes)
on exit:
AL = 1Bh
ES:DI = pointer to buffer with functionality and state information
[DI+00h] word = offset to static functionality information
[DI+02h] word = segment to static functionality information
[DI+04h] byte = current video mode
[DI+05h] word = character columns on screen
[DI+07h] word = page size in number of bytes
[DI+09h] word = starting address of current page
[DI+0Bh] word = cursor position for eight display pages
[DI+1Bh] word = current cursor type
[DI+1Dh] byte = current active page
[DI+1Eh] word = current CRTC address
[DI+20h] byte = current 3x8 register setting
[DI+21h] byte = current 3x9 register setting
[DI+22h] byte = number of character rows on screen
[DI+23h] word = number of scan lines per character
[DI+25h] byte = active display combination code
[DI+26h] byte = alternate display combination code
[DI+27h] word = number of colors supported in current mode
[DI+29h] byte = number of pages supported in current mode
[DI+2Ah] byte = 0 ; 200 scan lines in current mode
= 1 ; 350 scan lines in current mode
= 2 ; 400 scan lines in current mode
= 3 ; 480 scan lines in current mode
[DI+2Bh] byte = *Reserved*
[DI+2Ch] byte = *Reserved*
[DI+2Dh] byte = miscellaneous state information
bits 7, 6 = *Reserved*
bit 5 = 0 ; background intensity
= 1 ; blinking
bit 4 = 1 ; cursor emulation active
bit 3 = 1 ; mode set default palette loading disabled
bit 2 = 1 ; monochrome display attached
bit 1 = 1 ; summing active
bit 0 = 1 ; all modes on all display active
[DI+2Eh] byte = *Reserved*
[DI+2Fh] byte = *Reserved*
[DI+30h] byte = *Reserved*
[DI+31h] byte = 3; 256Kb of video memory available
[DI+32h] byte = save pointer information
bits 7, 6 = *Reserved*
bit 5 = 1 ; DCC extension active
bit 4 = 1 ; palette override active
bit 3 = 1 ; graphics font override active
bit 2 = 1 ; alpha font override active
bit 1 = 1 ; dynamic save area active

bit 0 = 1 ; 512 character set active
[DI+33h] 13 bytes = *Reserved*

static functionality table format
0 - function not supported
1 - supported function

[00h] byte = supported video mode
bit 7 = mode 07h
bit 6 = mode 06h
bit 5 = mode 05h
bit 4 = mode 04h
bit 3 = mode 03h
bit 2 = mode 02h
bit 1 = mode 01h
bit 0 = mode 00h
[01h] byte = supported video mode
bit 7 = mode 0Fh
bit 6 = mode 0Eh
bit 5 = mode 0Dh
bit 4 = mode 0Ch
bit 3 = mode 0Bh
bit 2 = mode 0Ah
bit 1 = mode 09h
bit 0 = mode 08h
[02h] byte = supported video mode
bits 7 to 4= *Reserved*
bit 3 = mode 13h
bit 2 = mode 12h
bit 1 = mode 11h
bit 0 = mode 10h
[03h] to [06h] = *Reserved*
[07h] = scan lines availabe in text modes
bits 7 to 3 = *Reserved*
bit 2 = 400 scan lines
bit 1 = 350 scan lines
bit 0 = 200 scan lines
[08h] = number of character fonts available in text modes
[09h] = maximun number of character fonts that can be active in text modes
[0Ah] byte = miscellaneous functions
bit 7 = color paging
bit 6 = color palette (color register)
bit 5 = EGA palette
bit 4 = cursor emulation
bit 3 = default palette loading when mode set
bit 2 = character font loading
bit 1 = color palette summing
bit 0 = all modes supported on all displays
[0Bh] = scan lines availabe in text modes

bits 7 to 4 = *Reserved*
bit 3 = DCC supported
bit 2 = background intensity/blinking control
bit 1 = save/restore supported
bit 0 = light pen supported
[0Ch] to [0Dh] = *Reserved*
[0Eh] = save pointer fucntions
bits 7 to 6 = *Reserved*
bit 5 = DCC extension supported
bit 4 = palette override
bit 3 = graphics font override
bit 2 = alpha font override
bit 1 = dynamic save area
bit 0 = 512-character set
[0Fh] = *Reserved*

AH=1Ch ; save and restore video state

AL = 0 ; return video save state bufffer size requirement
CX = requested states
bit 0 = video hardware state
bit 1 = video BIOS data area
bit 2 = video DAC state and color registers
on exit:
AL = 1Ch
BX = number of 64 bytes block required for the states requested in CX
AL = 1 ; save video state
CX = requested states (see AL=0)
ES:BX=pointer to buffer to store the video states information
on exit:
AL = 1Ch
AL = 2 ; restore video state
CX = requested states (see AL=0)
ES:BX = pointer to buffer with previous saved video states information
on exit:
AL = 1Ch

This page intentionally left blank.

Index

Perfecting the PC

User Response Form

In our continuing effort to improve our products and documentation, ATI Technologies Inc. is anxious to obtain your feedback on this manual. Using a scale of 1 to 5, with 5 representing the most favorable, and 1 the least favorable, please circle the number that best reflects your opinion. Everyone who completes and returns this questionnaire will receive an ATI souvenir (e.g. golf shirt, mug etc.) depending on availability.

Name and Organization ____

Title ______________________________ **Years in position** __________

Name of Manual *mach64* VGA Register Guide________ **Release No.** ______________

How do you rate the manual, generally?	*1*	*2*	*3*	*4*	*5*
Is the technical level of the manual appropriate?	*1*	*2*	*3*	*4*	*5*
Are operating instructions clear and complete?	*1*	*2*	*3*	*4*	*5*
Are terms and concepts explained clearly?	*1*	*2*	*3*	*4*	*5*
Is the manual well organized?	*1*	*2*	*3*	*4*	*5*
Are the tables easy to understand?	*1*	*2*	*3*	*4*	*5*
Is the artwork clear and easy to understand?	*1*	*2*	*3*	*4*	*5*
How helpful is the index?	*1*	*2*	*3*	*4*	*5*
How does this manual compare with other, similar manuals you have used?	*1*	*2*	*3*	*4*	*5*

Comments

What did you like best about this manual ?________________________________

____

What did you like least about this manual ? _______________________________

____

Are there any errors/omissions in the manual?_____________________________

____

THANK YOU FOR YOUR COMMENTS! Please mail to: ATI Technologies Inc., 33 Commerce Valley Drive East, Thornhill, Ontario, Canada L3T 7N6. You may also fax this to (905) 882-2620 (attn: Technical Publications)

This page intentionally left blank

TECHNOLOGIES INC.

Perfecting the PC

Printed in Canada