
# VGA Function Specification

# GXm/MXi Processors

January 16, 1998 - Rev. 0.1

Cyrix Corporation Confidential

Cyrix Corporation
2703 North Central Expressway
Richardson, Texas 75080
United States of America

# Table of Contents

## Table of Contents

## Table of Contents

# 1 VGA Function Overview

## 1.1 Introduction

The VGA graphics subsystem is compatible with the IBM VGA as defined in the IBM Video Subsystem Technical Reference manual. Additional functionality is provided to support the VESA BIOS Extensions (VBE) which provide higher screen addressability and color content than the base VGA alone. Support is also provided for a number of hardware-specific features.

## 1.2 VGA Modes

A VGA “mode” is a programmed VGA configuration defined by the VGA BIOS that produces a graphics frame buffer format and a screen image with specific characteristics. The base VGA function provides coded text modes for text-based applications, and graphics modes for graphics-based applications. Many of these modes are compatible with older graphics adapter standards, such as MDA, CGA, and EGA.

### 1.2.1 Text Modes

There are five text modes defined by VGA BIOS as shown in Table 1-1. Each of the text modes provides a coded frame buffer consisting of a 16-bit value for each character. The low byte is the ASCII character code for the character to display, and the high byte is an attribute byte that determines how the character is displayed (foreground, background colors, blink, underline, etc.). There are two formats defined by BIOS for the attribute byte: color and monochrome as shown in Table 1-2.

**Table 1-1 VGA Text Modes**

| BIOS Mode # | Screen Size in Characters | Attribute Type | Buffer Address | Compatibility |
|---|---|---|---|---|
| 0, 1 | 40x25 | Color | B8000-BFFFF | CGA |
| 2, 3 | 80x25 | Color | B8000-BFFFF | EGA, VGA |
| 7 | 80x25 | Monochrome | B0000-B7FFF | MDA |

**Table 1-2 Text Mode Attribute Byte Format**

| Bit | Color Definition | Monochrome Definition |
|---|---|---|
| 7 | Blink | Blink |
| 6 | Background Color (R) | Background |
| 5 | Background Color (G) | Background |
| 4 | Background Color (B) | Background |
| 3 | Foreground Intensity/Font Select | Foreground Intensity/Font Select |
| 2 | Foreground Color (R) | Foreground |
| 1 | Foreground Color (G) | Foreground |
| 0 | Foreground Color (B) | Underline |

### 1.2.2 Graphics Modes

The graphics modes listed in Table 1-3 are defined by VGA BIOS.

### 1.2.3 Major Components

VGA consists of several major components. These are:

- *Frame Buffer* - 256KB of memory organized as 64K DWORDS. It is viewed by the CPU as a flat buffer or as four planes (or maps) of 64K bytes each depending on the mode programmed.
- *Sequencer* - Controls clocking and converting multi-byte CPU accesses into discrete byte accesses for the Graphics Controller.
- *Graphics Controller* - Provides frame buffer manipulation assistance for the CPU. Frame buffer data can be read and written in various formats. Functions such as ROP, masking, and color compare are available.
- *Attribute Controller* - Formats the data read from the frame buffer for display on the screen.
- *General Registers* - Provide various control functions such as memory and I/O space location.
- *Video DAC* - Converts the digital pixel values to the analog RGB signals used by the display. It also contains the color lookup table.
- *CRT Controller* - Generates frame buffer address for display refresh, and display timing signals.

**Table 1-3 VGA Graphics Modes**

| BIOS Mode # | Screen Size in Pixels | # of Colors | Frame Buffer Format | Buffer Address |
|---|---|---|---|---|
| 4, 5 | 320x200 | 4 | Packed Pixel | B8000-BFFFF |
| 6 | 640x200 | 2 | Packed Pixel | B8000-BFFFF |
| 0xD | 320x200 | 16 | Planar | A0000-AFFFF |
| 0xE | 640x200 | 16 | Planar | A0000-AFFFF |
| 0xF | 640x400 | 4 | Planar | A0000-AFFFF |
| 0x10 | 640x350 | 16 | Planar | A0000-AFFFF |
| 0x11 | 640x480 | 2 | Planar | A0000-AFFFF |
| 0x12 | 640x480 | 16 | Planar | A0000-AFFFF |
| 0x13 | 320x200 | 256 | Packed Pixel | A0000-AFFFF |

# 2 VGA Operation

This chapter provides more details of the operation of the various components of the VGA subsystem. Each of the seven major components mentioned in the previous chapter is described in detail.

## 2.1 Frame Buffer

The VGA frame buffer consists of 256KB of memory organized as 64K DWORDS. Each byte in the DWORD is treated as a separate map (i.e. bytes 0, 1, 2, &3 are maps 0, 1, 2, & 3 respectively). In planar graphics modes, the maps correspond to the red, green, blue, and intensity color planes.

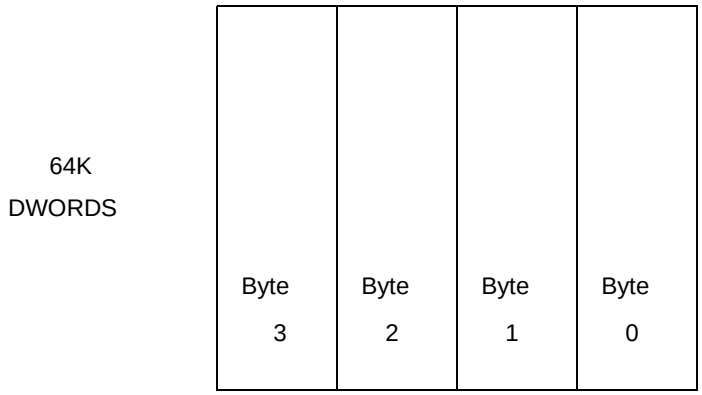
The CPU and the CRT refresh activity view the frame buffer in different ways depending on the setting of various VGA control bits. These two views are independent of each other. The CPU could be viewing the frame buffer in one way, and the CRT in another. The video BIOS modes set the two views to be (somewhat) consistent with each other, but it is possible for software to program the two in different ways (many games do this).

## 2.2 Sequencer

The Sequencer provides clocking to the other VGA sections. It generates the dot clock, character clock, and the shift/load signals used to control the operation of the video serializers. It also manages part of the CPU accesses to video memory.

The Map Mask register in this section enables or disabled CPU writes to each of the 4 memory maps.

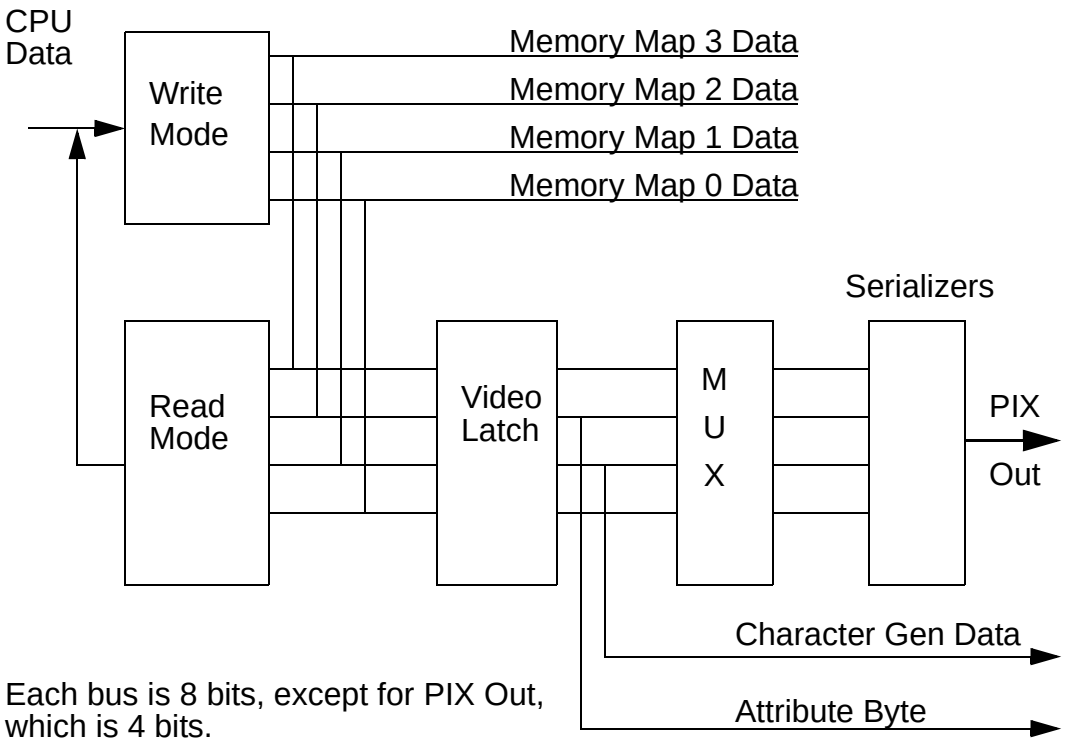
**Figure 2-1 Frame Buffer Organization**



## 2.3 Graphics Controller

The Graphics Controller manages the CPU interaction with video memory, and contains the video serializers that feed the front end of the Attribute Controller. Several memory read and write modes are supported that provide various forms of acceleration for VGA graphics operations. A high-level diagram of the Graphics Controller is shown below.

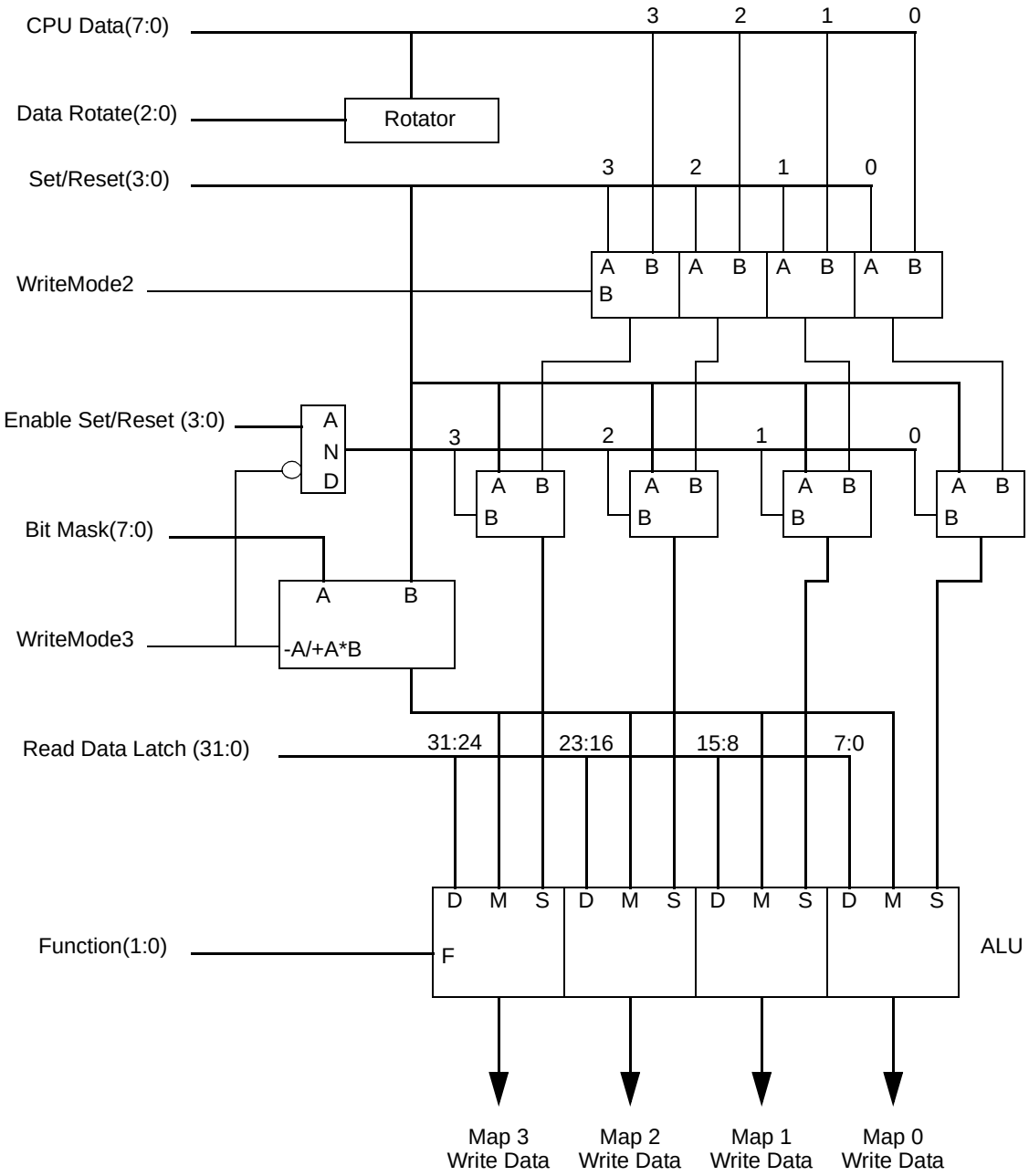
**Figure 2-2 Graphics Controller High-Level Diagram**



### 2.3.1 Write Modes

There are four write modes supported by the Graphics Controller (Mode 0, 1, 2, & 3). These write modes provide assistance to the CPU when the frame buffer is in a planar graphics format. The diagram below shows the data flow logic that supports these modes.

**Figure 2-3 Write Mode Data Flow**



### 2.3.2 Read Modes

There are two read modes provided to assist the CPU with graphics operations in planar modes. Read mode 0 simply returns the frame buffer data. Read mode 1 allows the CPU to do a single color compare across 8 pixels. Figure 2-4 shows the data flow for read modes. Figure 2-5 shows how the Color Compare block works in the previous figure.

**Figure 2-4 Read Mode Data Flow**

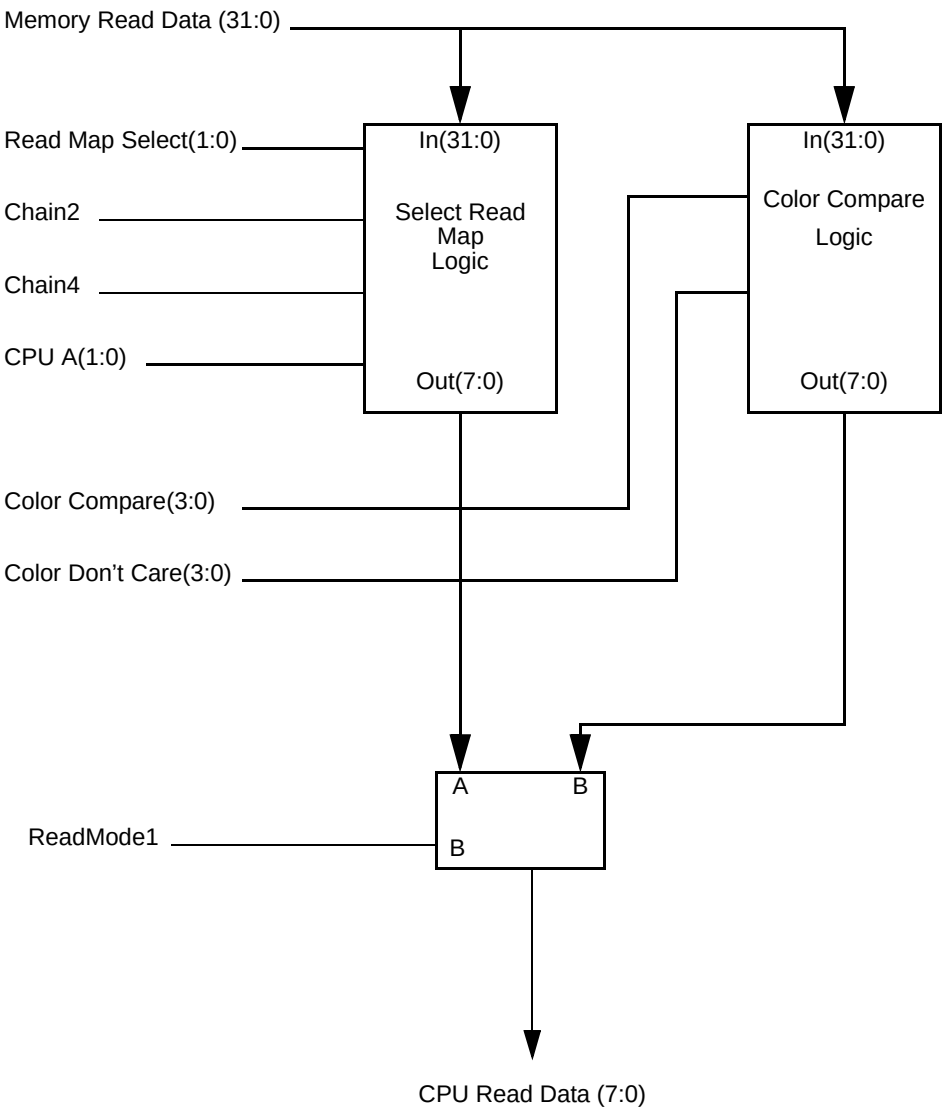
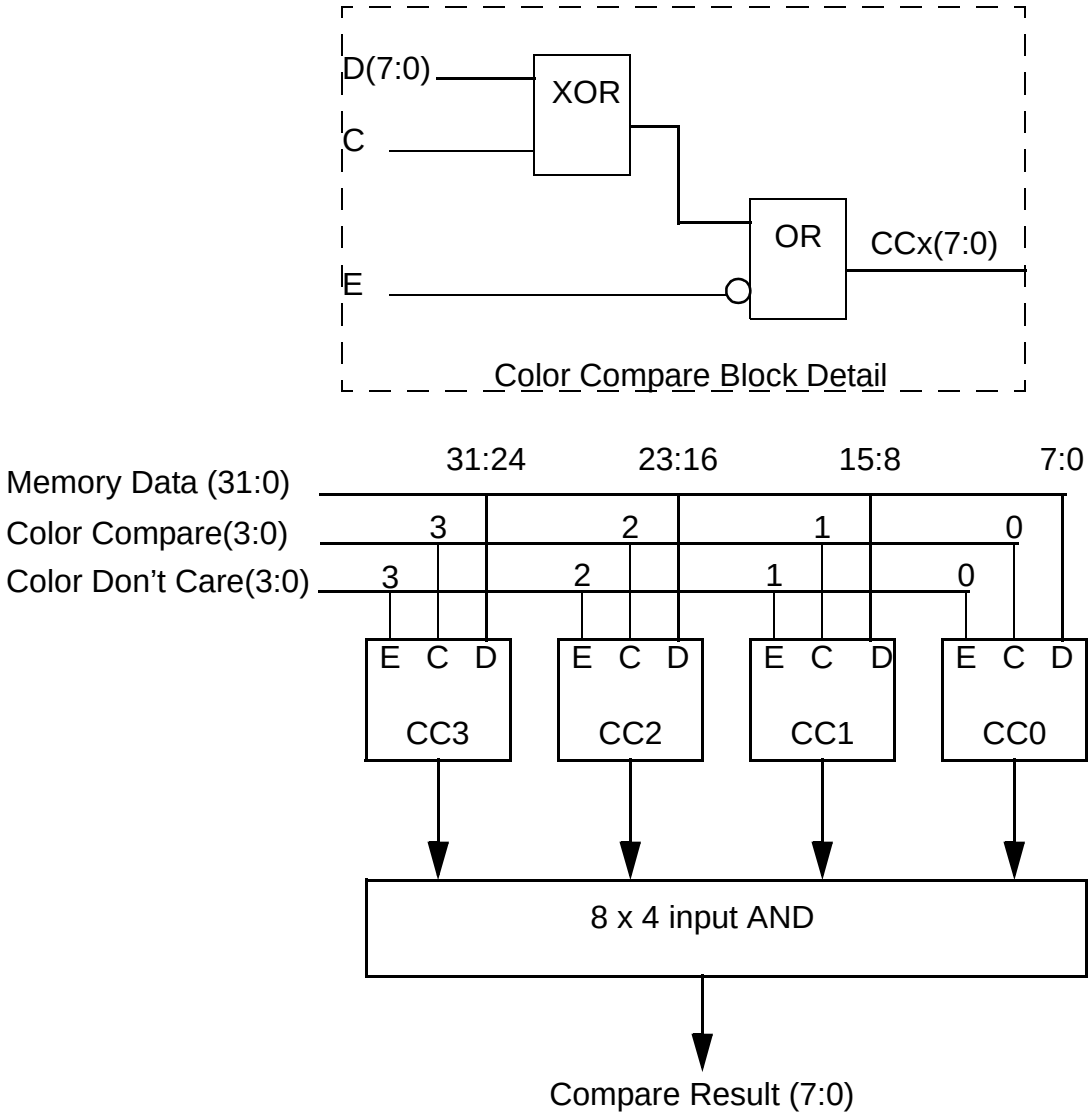


**Figure 2-5 Color Compare Operation**



## 2.4 Attribute Controller

The Attribute Controller takes serialized video memory data from the serializers in the Graphics Controller and formats it to be sent to the video DAC. In text modes, the character bit pattern is operated on by the attribute byte value.

## 2.5 General Registers

The General Registers contain miscellaneous registers that control general parts of the VGA, such as address decoding. Several status registers provide information about the video subsystem state.

## 2.6 Video DAC

The Video DAC takes an 8-bit pixel stream from the Attribute Controller and uses it as an index into the 256-location color lookup table (CLUT). Each entry in the CLUT is an 18-bit value containing 6 bits for each of the red, green, and blue color components. The appropriate color value is then converted to an analog signal by a high-speed DAC. The analog RGB signals are then sent to the display, along with the sync signals from the CRT Controller.

## 2.7 CRT Controller

The CRT Controller is responsible for generating the timing signals for the display (horizontal sync, vertical sync, blanking, etc.). It also initiates the display refresh memory reads that provide data to the video serializers in the Graphics Controller.

# 3 Registers

**Table 3-1 VGA Register I/O Addresses**

| Register/ Group | I/O Read Address | I/O Write Address | Comments |
|---|---|---|---|
| Miscellaneous Output Register | 0x3CC | 0x3C2 (W) | |
| Input Status Register 0 | 0x3C2 | - | |
| Input Status Register 1 | 0x3?A | - | ? = B (monochrome) or D (color) |
| Feature Control Register | 0x3CA | 0x3?A | ? = B (monochrome) or D (color) |
| Sequencer Index Register | 0x3C4 | 0x3C4 | |
| Sequencer Data Register | 0x3C5 | 0x3C5 | |
| CRT Controller Index Register | 0x3?4 | 0x3?4 | ? = B (monochrome) or D (color) |
| CRT Controller Data Register | 0x3?5 | 0x3?5 | ? = B (monochrome) or D (color) |
| Graphics Controller Index Reg. | 0x3CE | 0x3CE | |
| Graphics Controller Data Reg. | 0x3CF | 0x3CF | |
| Attribute Controller Index Reg. | 0x3C0 | 0x3C0 | |
| Attribute Controller Data Reg. | 0x3C1 | 0x3C0 | |
| Video DAC Index (CLUT Writes) | 0x3C8 | 0x3C8 | |
| Video DAC Index (CLUT Reads) | - | 0x3C7 | |
| Video DAC Data Register | 0x3C9 | 0x3C9 | |
| Video DAC Pel Mask Register | 0x3C6 | 0x3C6 | |

**Note:** The '?' in the I/O addresses above is determined by bit 0 of the Miscellaneous Output Register. See the description of this register for more information.

The following sections cover the six groups of registers that make up the VGA register set.

## 3.1 General Registers

### 3.1.1 Miscellaneous Output Register

*Read Address:* 0x3CC
*Write Address:* 0x3C2

| Bits | Function |
|---|---|
| 7 | **Vertical Sync Polarity** — Selects a positive-going VSYNC pulse (bit=0) or a negative-going VSYNC pulse (bit=1). |
| 6 | **Horizontal Sync Polarity** — Selects a positive-going HSYNC pulse (bit=0) or a negative-going HSYNC pulse (bit=1). |
| 5 | **Page Bit** — This bit is used to replace memory address bit A0 as the LSB when bit 1 of the Graphics Controller Miscellaneous Register is set to 1. |
| 4 | **Reserved** |
| 3:2 | **Clock Select** — Selects the VGA pixel clock source:<br>00 = Selects clock for 640/320 pixels per line<br>01 = Selects clock for 720/360 pixels per line<br>10 = Reserved<br>11 = Reserved |
| 1 | **RAM Enable** — Enables the video frame buffer address decode when set to 1. |
| 0 | **I/O Address Select** — Determines the I/O address of the CRTC registers (0x3?4 & 0x3?5), Feature Control Register (0x3?A), and Input Status Register 1 (0x3?A) as follows: ?=B when bit set to 0 (MDA I/O address emulation), ?=D when bit set to 1 (CGA address emulation). |

### 3.1.2 Input Status Register 0

*Read Address:* 0x3CC

| Bits | Function |
|---|---|
| 7 | **CRTC Interrupt Pending** — When a 1, indicates that a CRTC interrupt is pending. |
| 6:5 | Reserved |
| 4 | **Display Sense** — Returns the state of the wire AND of the RGB signal comparator outputs. This is used for display type sensing and diagnostics. |
| 3:0 | **Reserved** |

### 3.1.3 Input Status Register 1

*Read Address:* 0x3BA or 0x3DA

| Bits | Function |
|---|---|
| 7:4 | **Reserved** |
| 3 | **Vertical SYNC** — When a 1, indicates that the VSYNC signal is active. |
| 2:1 | **Reserved** |
| 0 | **Display Enable** — Reads as a 0 when both horizontal and vertical display enable are active. Reads as a 1 when either display enable signal is inactive. |

### 3.1.4 Feature Control Register

*Write Address:* 0x3BA or 0x3DA
*Read Address:* 0x3CA

| Bits | Function |
|---|---|
| 7:0 | **Reserved** |

## 3.2 Sequencer Registers

*Index Register Address:* 0x3C4
*Data Register Address:* 0x3C5

The Sequencer registers are accessed by writing an index value to the Sequencer Index Register (0x3C4) and reading or writing the register using the Sequencer Data Register (0x3C5).

### 3.2.1 Sequencer Registers Summary

| Index | Register |
|---|---|
| - | Sequencer Index |
| 0 | Reset |
| 1 | Clocking Mode |
| 2 | Map Mask |
| 3 | Character Map Select |
| 4 | Memory Mode |

### 3.2.2 Sequencer Index Register

| Bits | Function |
|---|---|
| 7:3 | **Reserved** |
| 2:0 | **Index** |

### 3.2.3 Reset Register (Index 0)

| Bits | Function |
|---|---|
| 7:2 | **Reserved** |
| 1:0 | **Enable Display** — Both these bits should be set to 1 (value=11b) to enable display of the VGA screen image. If either of these bits are 0, the display will be blanked. The VGA will continue to respond to I/O and memory accesses. |

### 3.2.4 Clocking Mode (Index 1)

| Bits | Function |
|---|---|
| 7:6 | **Reserved** |
| 5 | **Screen Off** — Setting this bit to a 1 will blank the screen while maintaining the HSYNC and VSYNC signals. This is intended to allow the CPU full access to the memory bandwidth. This bit must be 0 for the display image to be visible. |
| 4 | **Not Supported ** (Shift4) |
| 3 | **Dot Clock by 2** — When set to 1, the incoming pixel clock will be divided by 2 to form the actual dot clock. When 0, the incoming pixel clock will be used unchanged. |
| 2 | **Not Supported** (Shift Load) |
| 1 | **Always 1** |
| 0 | **8-Dot Character Width** — When set to a 1, the character cells in text mode will be 8 pixels wide. When set to 0, the character cells will be 9 pixels wide. The 9th pixel will be equal to the 8th pixel for character codes 0xC0-0xDF (the line graphics character codes), and will be 0 (background) for all other codes. |

### 3.2.5 Map Mask (Index 2)

*Enable Map 3 (bit 3), Enable Map 2 (bit 2), Enable Map 1 (bit 1), Enable Map 0 (bit 0)*

These bits enable (bit = 1) writing to their corresponding bytes in each DWORD of the frame buffer. (i.e. EM3 enables byte 3, EM2 enables byte 2, etc.) The 4 maps or planes correspond to the 4 bytes in each DWORD of the frame buffer. Reads to all maps are always enabled, and are unaffected by these bits.

| Bits | Function |
|---|---|
| 7:4 | **Reserved** |
| 3 | **Enable Map 3** |
| 2 | **Enable Map 2** |
| 1 | **Enable Map 1** |
| 0 | **Enable Map 0** |

### Table 3-2 Character Map Select (Index 3)

*Character Map A (bits 5,3:2) and Character Map B (bits 4, 1:0)*

These fields determine which font tables are used when displaying a character in text mode. When attribute bit 3 = 1, Character Map A is used. When attribute bit 3 is a 0, Character Map B is used. The font tables are stored in the 64KB in map 2. There are 8 font tables. The character map codes select the font tables:

| Code | Font Table Location in Map 2 |
|---|---|
| 0 | 8KB block 0 |
| 1 | 8KB block 2 |
| 2 | 8KB block 4 |
| 3 | 8KB block 6 |
| 4 | 8KB block 1 |
| 5 | 8KB block 3 |
| 6 | 8KB block 5 |
| 7 | 8KB block 7 |

| Bits | Function |
|---|---|
| 7:6 | **Reserved** |
| 5 | Character Map A bit 2 |
| 4 | Character Map B bit 2 |
| 3:2 | Character Map A bits 1:0 |
| 1:0 | Character Map B bits 1:0 |

## 3.2.6 Memory Mode (Index 4)

| Bits | Function |
|---|---|
| 7:4 | **Reserved** |
| 3 | **Chain4** — When set to a 1, CPU address bits 1 and 0 are used to select the map or plane in the frame buffer DWORD. For example, if CPU A1:A0 = 3, then map 3 is selected. If CPU A1:A0=1, then map 1 is selected. If Chain4 is 0, then the frame buffer addressing is controlled by the Chain2 bit. |
| 2 | **Chain2 (was Odd/Even)** — When set to a 1, CPU address bit 0 selects between frame buffer maps 0 and 1, or maps 2 and 3 depending on the value in the Graphics Controller Read Map Select Register. For example, if CPU A0 is 0, then map 0 (or 2) is selected. |
| 1 | **Extended Memory** — This bit should always be set to a 1. It is a throwback to EGA where the standard frame buffer size was 64KB and was upgradeable to 256KB. VGA always has (at least) 256KB. |
| 0 | **Reserved** |

## 3.3 CRT Controller Registers

*Index Register Address:* 0x3B4 or 0x3D4
*Data Register Address:* 0x3B5 or 0x3D5

The CRTC registers are accessed by writing an index value to the CRTC Index Register (0x3B4 or 0x3D4) and reading or writing the register using the CRTC Data Register (0x3B5 or 0x3D5). See the description of the I/O Address Select bit in the Miscellaneous Output Register for more information on the I/O address of the CRTC registers.

Note that the Extended VGA Registers are accessed through the CRTC interface. This section only discusses the base VGA registers, however. See the Extended VGA Registers chapter for more information on the extended registers.

**Table 3-3 CRTC Registers Summary**

| Index | Register |
|---|---|
| - | CRTC Index |
| 0x00 | Horizontal Total |
| 0x01 | Horizontal Display Enable End |
| 0x02 | Horizontal Blank Start |
| 0x03 | Horizontal Blank End |
| 0x04 | Horizontal Sync Start |
| 0x05 | Horizontal Sync End |
| 0x06 | Vertical Total |
| 0x07 | Overflow |
| 0x08 | Preset Row Scan |
| 0x09 | Maximum Scan Line |
| 0x0A | Cursor Start |
| 0x0B | Cursor End |
| 0x0C | Start Address High |
| 0x0D | Start Address Low |
| 0x0E | Cursor Location High |
| 0x0F | Cursor Location Low |
| 0x10 | Vertical Sync Start |
| 0x11 | Vertical Sync End |
| 0x12 | Vertical Display Enable End |
| 0x13 | Offset |
| 0x14 | Underline Location |
| 0x15 | Vertical Blank Start |
| 0x16 | Vertical Blank End |
| 0x17 | CRTC Mode Control |
| 0x18 | Line Compare |
| 0x22 | CPU Data Latch State |
| 0x24 | Attribute Index/Data FF State |
| 0x26 | Attribute Index State |

### 3.3.1 CRTC Index Register

| Bits | Function |
|---|---|
| 7 | **Reserved** |
| 6:0 | **Index** |

### 3.3.2 Horizontal Total (Index 0)

| Bits | Function |
|---|---|
| 7:0 | **Horizontal Total** — This value specifies the number of character clocks per horizontal scanline minus 5. It determines the horizontal line rate/period. |

### 3.3.3 Horizontal Display Enable End (Index 1)

| Bits | Function |
|---|---|
| 7:0 | **Horizontal Display Enable End** — This value specifies the number of displayed characters minus 1. It determines the width of the Horizontal Display Enable signal. |

### 3.3.4 Horizontal Blank Start (Index 2)

| Bits | Function |
|---|---|
| 7:0 | **Horizontal Blank Start** — This value specifies the character position on the line where the horizontal blanking signal goes active. |

### 3.3.5 Horizontal Blank End (Index 3)

| Bits | Function |
|---|---|
| 7 | Set to 1 |
| 6:5 | **Display Enable Skew Control** — This value is a binary encoded value that specifies how many character clocks are needed to skew the horizontal display enable signal by (0 character clocks - 3 character clocks) before it is sent to the Attribute Controller. This field is used to accommodate differences in the length of the video pipeline (frame buffer to pixel output) in various text and graphics modes. |
| 4:0 | **Horizontal Blank End 4:0** — This 6-bit value is a compare target for the character count where the horizontal blank signal will end. Bit 5 of this value is in the Horizontal Sync End Register (bit 7). Note that not all horizontal counter bits are compared, which can create aliased compares depending upon the binary values involved in the count range and compare values. |

### 3.3.6 Horizontal Sync Start (Index 4)

| Bits | Function |
|---|---|
| 7:0 | **Horizontal Sync Start** — This value specifies the character position where the horizontal sync pulse will start. |

### 3.3.7 Horizontal Sync End (Index 5)

| Bits | Function |
|---|---|
| 7 | **Horizontal Blank End bit 5** — See the description of the Horizontal Blank End Register (index 5). |
| 6:5 | **Not Implemented** (HSync Delay) |
| 4:0 | **Horizontal Sync End** — These bits represent the low 5 bits of the character position where the Horizontal Sync signal ends. |

### 3.3.8 Vertical Total (Index 6)

| Bits | Function |
|---|---|
| 7:0 | **Vertical Total bits 7:0** — This represents the low 8 bits of a value that specifies the total number of scanlines on the screen minus 2. This value includes the blanking area and determines the vertical refresh rate. The high 2 bits of this value are in the Overflow Register (Index 7). |

### 3.3.9 Overflow (Index 7)

These are the high-order bits for several of the vertical programming values. See the descriptions of the respective vertical registers for descriptions of these fields.

| Bits | Function |
|---|---|
| 7 | **Vertical Sync Start bit 9** |
| 6 | **Vertical Display Enable End bit 9** |
| 5 | **Vertical Total bit 9** |
| 4 | **Line Compare bit 8** |
| 3 | **Vertical blank Start bit 8** |
| 2 | **Vertical Sync Start bit 8** |
| 1 | **Vertical Display Enable End bit 8** |
| 0 | **Vertical Total bit 8** |

### 3.3.10 Preset Row Scan (Index 8)

| Bits | Function |
|---|---|
| 7 | **Reserved** |
| 6:5 | **Byte Panning** — This value causes the pixel data stream to be fetched 0, 1, 2, or 3 character positions early for use with pel panning in the Attribute Controller. This field is used when the video serializers are chained together (by 2 or by 4). |
| 4:0 | **Starting Row Scan** — This specifies the value loaded into the row scan counter on the first text line of the screen. Changing this value in text modes allows the screen to be scrolled on a scanline basis rather than a text line basis. The starting row scan count for all subsequent scanlines is 0. |

### 3.3.11 Maximum Scan Line (Index 9)

| Bits | Function |
|---|---|
| 7 | **Double Scan** — When this bit is set to a 1, the row scan counter increments every other scan line. When this bit is cleared to 0, the row scan counter increments on every scan line. This bit is used to make 200 line text modes occupy 400 physical scan lines on the screen. |
| 6 | **Line Compare bit 9** — See the description of the Line Compare Register (Index 0x18) for a description of this field. |
| 5 | **Vertical Blank Start bit 9** — See the description of the Vertical Blank Start Register (Index 0x15) for a description of this field. |
| 4:0 | **Maximum Scan Line** — This field specifies the number of scan lines per character row minus 1. The row scan counter will count up to this value then go to 0 for the next character row. |

### 3.3.12 Cursor Start (Index 0xA)

| Bits | Function |
|---|---|
| 7:6 | **Reserved** |
| 5 | **Cursor Off** — When set to 1, the cursor will be turned off and will not appear on the screen. When this bit is 0, the cursor will be displayed. This bit is only applicable in text modes. |
| 4:0 | **Cursor Start** — This field specifies the first scanline in the character box where the cursor is displayed. If this value is greater than the Cursor End value, then no cursor will be displayed. If this value is equal to the Cursor End value, then the cursor occupies a single scan line. |

### 3.3.13 Cursor End (Index 0xB)

| Bits | Function |
|---|---|
| 7 | **Reserved** |
| 6:5 | **Cursor Skew** — This field allows the cursor to be skewed by 0, 1, 2, or 3 character positions to the right. |
| 4:0 | **Cursor End** — This field specifies the last scan line in the character box where the cursor is displayed. See the description of the Cursor Start Register (Index 0xA) for more information. |

### Start Address High (Index 0xC) and Start Address Low (Index 0xD)

Start Address

This value specifies the frame buffer address used at the beginning of a screen refresh. It represents the upper left corner of the screen.

| Bits | Function |
|---|---|
| **Start Address High (Index 0xC)** | |
| 7:0 | **Start Address bits 15:8** |
| **Start Address Low (Index 0xD)** | |
| 7:0 | **Start Address bits 7:0** |

### 3.3.14 Cursor Location High (Index 0xE) and Cursor Location Low (Index 0xF)

Cursor Location

This value specifies the frame buffer address where the cursor will be displayed in text mode. The cursor will appear at the character whose memory address corresponds to this value.

| Bits | Function |
|---|---|
| **Cursor Location High (Index 0xE)** | |
| 7:0 | Cursor Location bits 15:8 |
| **Cursor Location Low (Index 0xF)** | |
| 7:0 | Cursor Location bits 7:0 |

### 3.3.15 Vertical Sync Start (Index 0x10)

| Bits | Function |
|---|---|
| 7:0 | **Vertical Sync Start bits 7:0** — This value specifies the scanline number where the vertical sync signal will go active. This is a 10-bit value. Bits 8 and 9 are in the Overflow Register (Index 0x8). |

### 3.3.16 Vertical Sync End (Index 0x11)

| Bits | Function |
|---|---|
| 7 | **Write-protect Registers 0-7** — This bit is used to prevent old EGA programs from writing invalid values to the VGA horizontal timing registers. The Line Compare bit in the Overflow Register (Index 7) is not protected by this bit. |
| 6 | **Not Implemented** (Refresh Cycle Select) |
| 5 | **Not Implemented** (Enable Vertical Interrupt) |
| 4 | **Not Implemented** (Clear Vertical Interrupt) |
| 3:0 | **Vertical Sync End** — This field represents the low 4 bits of a compare value that specifies which scan line that the vertical sync signal goes inactive. |

### 3.3.17 Vertical Display Enable End (Index 0x12)

| Bits | Function |
|---|---|
| 7:0 | **Vertical Display Enable End bits 7:0** — This is a 10-bit value that specifies the scan line where the vertical display enable signal goes inactive. It represents the number of active scan lines minus 1. Bits 8 and 9 of this value are in the Overflow Register (index 7). |

### 3.3.18 Offset (Index 0x13)

| Bits | Function |
|---|---|
| 7:0 | **Offset** — This field specifies the logical line width of the screen. This value (multiplied by 2 or 4 depending on the CRTC clocking mode) is added to the starting address of the current scan line to get the starting address of the next scan line. |

### 3.3.19 Underline Location (Index 0x14)

| Bits | Function |
|---|---|
| 7 | **Reserved** |
| 6 | **Doubleword Mode** — When this bit is a 1, CRTC memory addresses are DWORD addresses, and the CRTC refresh counter effectively increments by 4. When this bit is a 0, the address increment is determined by the Byte Mode bit in the CRTC Mode Control Register (Index 0x17). See the description of the CRTC Mode Control Register (index 0x17) for more information. |
| 5 | **Count by 4** — When this bit is a 1, the CRT memory address counter is clocked with the character clock divided by 4. This is used with Doubleword mode. |
| 4:0 | **Underline Location** — This field specifies the row scan value where the underline will appear in the character box in text modes. |

### 3.3.20 Vertical Blank Start (Index 0x15)

| Bits | Function |
|---|---|
| 7:0 | **Vertical Blank Start bits 7:0** — This the low 8 bits of a value that specifies the starting scan line of the vertical blank signal. This is a 10-bit value. Bit 8 is in the Overflow Register (Index 7) and bit 9 is in the Maximum Scan Line Register (index 9). |

### 3.3.21 Vertical Blank End (Index 0x16)

| Bits | Function |
|---|---|
| 7:0 | **Vertical Blank End bits 7:0** — This value specifies the low 8 bits of a compare value that represents the scan line where the vertical blank signal goes inactive. |

### 3.3.22 CRTC Mode Control (Index 0x17)

| Bits | Function |
|---|---|
| 7 | **Enable Syncs** — When set to 1, this bit will enable the horizontal and vertical sync signals. When 0, this bit will hold both sync flip-flops reset. |
| 6 | **Byte Mode** — If the Doubleword Mode bit in the Underline Location register is 0, then this bit will configure the CRTC addresses for byte addresses (Byte Mode=1) or word addresses (Byte Mode=0). If the Doubleword Mode bit is a 1, then the Byte Mode bit is ignored. See the diagram at the end of this section for more information on the various CRTC addressing modes. |
| 5 | **Address Wrap** — When the CRTC is addressing the frame buffer in Word Mode (Byte Mode=0, Doubleword Mode=0) then this bit determines which address bit occupies the MA0 bit position of the address sent to the frame buffer memory. If Address Wrap=0, CRTC address counter bit 13 occupies the MA0 position. If Address Wrap=1, then CRTC address counter bit 15 is in the MA0 position. See the diagram at the end of this section for more information on the various CRTC addressing modes. |
| 4 | **Reserved** |
| 3 | **Count by 2** — When this bit is a 1, the CRTC address counters are clocked by the character clock divided by 2. When this bit is a 0, the CRTC address counters are clocked by the normal character clock. This bit is overridden (a don't care) if the Count by 4 bit in the Underline Location Register (index 0x14) is a 1. |
| 2 | **VCLK Select** — This bit determines the clocking for the vertical portion of the CRTC. If this bit is 0, the horizontal sync signal clocks the vertical section. If this bit is 1, the horizontal sync divided by 2 clocks the vertical section. |
| 1 | **Select Row Scan Bit** — This bit determines which CRTC signal appears on the MA14 address bit sent to the frame buffer memory. If this bit is a 0, bit 1 of the Row Scan counter appears on MA14. If this bit is a 1, then CRTC address counter bit 14, 13, or 12 appears on MA14. See the table at the end of this section for more information. |
| 0 | **Select A13** — This bit determines which CRTC signal appears on the MA13 address bit sent to the frame buffer memory. If this bit is a 0, bit 0 of the Row Scan counter appears on MA13. If this bit is a 1, then CRTC address counter bit 13, 12, or 11 appear on MA13. See the table at the end of this section for more information. |

Table 3-4 illustrates the various frame buffer addressing schemes. In the table, MAx represents the frame buffer memory address signals, Ax represents the CRTC address counter signals, RSx represents row scan counter output bits. The binary value in the column headings is a concatenation of the Doubleword Mode and Byte Mode bits. (i.e. {DoublewordMode, ByteMode} in Verilog.)

**Table 3-4 CRTC Memory Addressing Modes**

| Frame Buffer Memory Address bit | Byte Mode (01) | Word Mode (00) | DWORD Mode (1X) |
|---|---|---|---|
| MA0 | A0 | A15 or A13 | A12 |
| MA1 | A1 | A0 | A13 |
| MA2 | A2 | A1 | A0 |
| MA3 | A3 | A2 | A1 |
| MA4 | A4 | A3 | A2 |
| MA5 | A5 | A4 | A3 |
| MA6 | A6 | A5 | A4 |
| MA7 | A7 | A6 | A5 |
| MA8 | A8 | A7 | A6 |
| MA9 | A9 | A8 | A7 |
| MA10 | A10 | A9 | A8 |
| MA11 | A11 | A10 | A9 |
| MA12 | A12 | A11 | A10 |
| MA13 | A13 or RS0 | A12 or RS0 | A11 or RS0 |
| MA14 | A14 or RS1 | A13 or RS1 | A12 or RS1 |
| MA15 | A15 | A14 | A13 |

### 3.3.23 Line Compare (Index 0x18)

| Bits | Function |
|---|---|
| 7:0 | **Line Compare bits 7:0** — This value specifies the low 8 bits of a compare value that represents the scan line where the CRTC frame buffer address counter is reset to 0. This can be used to create a split screen by using the Start Address registers to specify a non-zero location at which to begin the screen image. The lower portion of the screen (starting at frame buffer address 0) will be immune to screen scrolling (and pel panning as specified in the Attribute Controller Mode Control Register (index 0x10). Line Compare is a 10-bit value. Bit 8 is located in the Overflow register (index 7) and bit 9 is in the Maximum Scan Line register (index 9). |

### 3.3.24 CPU Data Latch State (Index 0x22)

| Bits | Function |
|---|---|
| 7:0 | **Data Latch Value** — This read-only field returns a byte of the CPU data latches and can be used in VGA save/restore operations. The Read Map Select field in the Graphics Controller specifies which byte/map (0-3) is returned. |

### 3.3.25 Attribute Index/Data FF State (Index 0x24)

| Bits | Function |
|---|---|
| 7 | **FF State** — This read-only bit indicates the state of the Attribute Controller index/data flip-flop. When this bit is 0, the next write to 0x3C0 will write an index value, when this bit is 1, the next write to 0x3C0 will write a data register value. |
| 6:0 | **Reserved** |

### 3.3.26 Attribute Index State (Index 0x26)

| Bits | Function |
|---|---|
| 7:6 | **Reserved** |
| 5:0 | **Attribute Index Value** — This read-only value indicates the value of Attribute Index Register bits 5:0. |

## 3.4 Graphics Controller Registers

*Index Register Address:* 0x3CE
*Data Register Address:* 0x3CF

The Graphics Controller registers are accessed by writing an index value to the Graphics Controller Index Register (0x3CE) and reading or writing the register using the Graphics Controller Data Register (0x3CF).

**Table 3-5 Graphics Controller Registers Summary**

| Index | Register |
|---|---|
| - | Graphics Controller Index |
| 0 | Set/Reset |
| 1 | Enable Set/Reset |
| 2 | Color Compare |
| 3 | Data Rotate |
| 4 | Read Map Select |
| 5 | Graphics Mode |
| 6 | Miscellaneous |
| 7 | Color Don't Care |
| 8 | Bit Mask |

### 3.4.1 Graphics Controller Index Register

| Bits | Function |
|---|---|
| 7:4 | **Reserved** |
| 3:0 | **Index** |

### 3.4.2 Set/Reset (Index 0)

| Bits | Function |
|---|---|
| 7:4 | **Reserved** |
| 3 | **Set/Reset Map 3** — Allows bits in Map 3 to be set or reset through write modes 0 or 3. See the description of the VGA write modes for more information. |
| 2 | **Set/Reset Map 2** — Allows bits in Map 2 to be set or reset through write modes 0 or 3. See the description of the VGA write modes for more information. |
| 1 | **Set/Reset Map 1** — Allows bits in Map 1 to be set or reset through write modes 0 or 3. See the description of the VGA write modes for more information. |
| 0 | **Set/Reset Map 0** — Allows bits in Map 0 to be set or reset through write modes 0 or 3. See the description of the VGA write modes for more information. |

### 3.4.3 Enable Set/Reset (Index 1)

| Bits | Function |
|---|---|
| 7:4 | **Reserved** |
| 3 | Enable Set/Reset Map 3 — Enables the Set/Reset function for Map 3 in write mode 0. See the description of the VGA write modes for more information. |
| 2 | **Enable Set/Reset Map 2** — Enables the Set/Reset function for Map 2 in write mode 0. See the description of the VGA write modes for more information. |
| 1 | **Enable Set/Reset Map 1** — Enables the Set/Reset function for Map 1 in write mode 0. See the description of the VGA write modes for more information. |
| 0 | **Enable Set/Reset Map 0** — Enables the Set/Reset function for Map 0 in write mode 0. See the description of the VGA write modes for more information. |

### 3.4.4 Color Compare (Index 2)

Color Compare Map 3 - 0 (bits 3 - 0)
These bits specify a compare value that allows the CPU to compare pixels in planar modes. Read mode 1 will perform a comparison based on these bits combined with the Color Don't Care bits. Data returned will contain a 1 in each one of the eight pixel positions where a color match is found. See the description of VGA read modes for more information.

| Bits | Function |
|---|---|
| 7:4 | **Reserved** |
| 3 | **Color Compare Map 3** |
| 2 | **Color Compare Map 2** |
| 1 | **Color Compare Map 1** |
| 0 | **Color Compare Map 0** |

### 3.4.5 Data Rotate (Index 3)

| Bits | Function |
|---|---|
| 7:5 | **Reserved** |
| 4:3 | **Write Operation** — Data written to the frame buffer by the CPU can be logically combined with data already in the CPU data latches.<br><br>00 = Copy (CPU data written unmodified)<br>01 = CPU data ANDed with Latched Data<br>10 = CPU data ORed with Latched Data<br>11 = CPU data XORed with Latched Data<br><br>See the description of VGA write modes for more information. |
| 2:0 | **Rotate Count** — This value is used to rotate the CPU data before it is used in write modes 0 and 3. The CPU data byte written is rotated right, with low bits wrapping to the high bit positions.<br><br>See the description of VGA write modes for more information. |

### 3.4.6 Read Map Select (Index 4)

| Bits | Function |
|---|---|
| 7:2 | **Reserved** |
| 1:0 | **Read Map Select** — This field specifies which map CPU read data is taken from in read mode 0. In Odd/Even modes (specified by the Odd/Even bit in the Graphics Mode Register) bit 1 of this field specifies which pair of maps returns data. When bit 1 is 0, data is returned from maps 0 & 1. When bit 1 is 1, data is returned from maps 2 & 3. The CPU read address bit A0 determines which byte is returned (low or high) in Odd/Even modes. In non-Odd/Even modes, both bits 1 & 0 of the Read Map Select field specify the map to read (0, 1, 2, or 3) and the CPU accesses data sequentially within the specified map. |

### Table 3-6 Graphics Mode (Index 5)

| Bits | Function |
|---|---|
| 7 | **Reserved** |
| 6 | **256 Color Mode** — When set to a 1, this bit configures the video serializers in the Graphics controller for the 256 color mode (BIOS mode 0x13). When this bit is 0, the Shift Register Mode bit controls the serializer configuration. |
| 5 | **Shift Register Mode** — When set to a 1, this bit configures the video serializers for BIOS modes 4 and 5. When this bit is 0, the serializers are taken in paralleled, i.e. configured for 4-bit planar mode operation. NOTE that the serializers are also wired together serially so that map 3 bit 7 feeds map 2 bit 0, map 2 bit 7 feeds map 1 bit 0, and map 1 bit 7 feeds map 0 bit 0. This allows for a 32-pixel 1 bit-per-pixel serializer to be used. For this configuration, color planes 1, 2, & 3 should be masked off using the Color Plane Enable register in the Attribute Controller. |
| 4 | **Odd/Even** — When this bit is set to 1, CPU address bit A0 will select between maps 0 & 1 or maps 2 & 3 depending on the state of the Read Map Select register. When this bit is 0, the CPU accesses data sequentially within a map. This bit is equivalent to the Odd/Even bit in the Sequencer Memory Mode Register, but is inverted in polarity from that bit. |
| 3 | **Read Mode** — This bit determines what is returned to the CPU when it reads the frame buffer. When this bit is 1, the result of a color compare operation is returned. The 8 bits in the CPU read data contain a 1 in each pixel position where the color compare operation was true, and a 0 where the operation was false. When this bit is 0, frame buffer map data is returned. |
| 2 | **Reserved** |
| 1:0 | **Write Mode** — This field specifies how CPU data is written to the frame buffer. Table 3-7 describes each write mode. Note that the Write Operation field in the Data Rotate Register specifies how CPU data is combined with data in the data latches for write modes 0, 2, and 3. |

### Table 3-7 Graphics Write Modes

| Write Mode | Function |
|---|---|
| 0 | CPU data is rotated by the count in the Data Rotate Register. Each map enabled by the Sequencer Map Mask Register is written by the rotated CPU data combined with the latch data (if set/reset is NOT enabled for that map) or by the map's corresponding set/reset bit replicated across the 8-bit byte (if set/reset IS enabled for that map). The Bit Mask Register is used to protect individual bits in each map from being updated. |
| 1 | Each map enabled by the Sequencer Map Mask Register is written with its corresponding byte in the data latches. |
| 2 | CPU data is replicated for each map and combined with the data latches and written to memory. The Bit Mask Register is used to protect individual bits in each map from being updated. |
| 3 | Each map is written with its corresponding Set/Reset bit replicated through a byte (Enable Set/Reset is ignored). The CPU data is rotated and ANDed with the Bit Mask Register. The resulting mask is used to protect individual bits in each map. |

### 3.4.7 Miscellaneous (Index 6)

| Bits | Function |
|---|---|
| 7:4 | **Reserved** |
| 3:2 | **Memory Map** — This field controls the address mapping of the frame buffer in the CPU memory space. The bits are encoded as follows:<br>00 = A0000 to BFFFF (128KB)<br>01 = A0000 to AFFFF (64KB)<br>10 = B0000 to B7FFF (32KB)<br>11 = B8000 to BFFFF (32KB) |
| 1 | **Odd/Even** — When set to 1, this bit replaces the CPU A0 address bit with a higher order bit when addressing the frame buffer. Odd maps are then selected when CPU A0=1, and even maps selected when CPU A0=0. |
| 0 | **Graphics Mode** — This bit specifies graphics mode (bit=1) or text mode (bit=0) operation. |

### 3.4.8 Color Don’t Care (Index 7)

| Bits | Function |
|---|---|
| 7:4 | **Reserved** |
| 3 | **Compare Map 3** — Enables (bit=1) or excludes (bit=0) Map 3 from participating in a color compare operation. |
| 2 | **Compare Map 2** — Enables (bit=1) or excludes (bit=0) Map 2 from participating in a color compare operation. |
| 1 | **Compare Map 1** — Enables (bit=1) or excludes (bit=0) Map 1 from participating in a color compare operation. |
| 0 | **Compare Map 0** — Enables (bit=1) or excludes (bit=0) Map 0 from participating in a color compare operation. |

### 3.4.9 Bit Mask (Index 8)

| Bits | Function |
|---|---|
| 7:0 | **Bit Mask** — The Bit Mask is used to enable or disable writing to individual bits in each map. A 1 in the bit mask allows a bit to be updated, while a 0 in the bit mask writes the contents of the data latches back to memory, effectively protecting that bit from update. The data latches must be set by doing a frame buffer read in order for the masking operation to work properly. The bit mask is used in write modes 0, 2, and 3. |

## 3.5 Attribute Controller Registers

*Index Register Address:* 0x3C0
*Data Register Address:* 0x3C0 (Write) 0x3C1 (Read)

The Attribute Controller registers are accessed by writing an index value to the Attribute Controller Index Register (0x3C0) and reading or writing the register using the Attribute Controller Data Register (0x3C0 for writes, 0x3C1 for reads).

**Table 3-8 Attribute Controller Registers Summary**

| Index | Register |
|---|---|
| - | Attribute Index |
| 0x00-0x0F | EGA Palette Registers |
| 0x10 | Attribute Mode Control |
| 0x11 | Overscan Color |
| 0x12 | Color Plane Enable |
| 0x13 | Horizontal Pel Panning |
| 0x14 | Color Select |

### 3.5.1 Attribute Controller Index Register

The Attribute Controller registers do not have a separate address for writing index and data information. Instead, an internal flip-flop alternates between index and data registers. Reading Input Status Register 1 (0x3BA or 0x3DA) clears the flip-flop to the index state. The first write to 0x3C0 following a read from Input Status Register 1 will update the index register. The next write will update the selected data register. The next write specifies a new index, etc.

| Bits | Function |
|---|---|
| 7:6 | **Reserved** |
| 5 | **Internal Palette Address** — This bit determines whether the EGA palette is addressed by the video pixel stream (bit=1) or by the Attribute Controller Index Register (bit=0). This bit should be set to 1 for normal VGA operation. CPU I/O accesses to the palette are disabled unless this bit is a 0. |
| 4:0 | **Data Register Index** — This field addresses the individual palette and data registers. |

### 3.5.2 EGA Palette (Indexes 0x00 - 0x0F)

| Bits | Function |
|---|---|
| 7:6 | **Reserved** |
| 5:0 | **Color Value** — Each of these 16 registers is used to expand the pixel value from the frame buffer (1, 2, or 4 bits wide) into a 6-bit color value that is sent to the video DAC. The EGA palette is “programmed out of the way” in 256 color mode. These registers can only be read or written when the Internal Palette Address bit in the Index register is 0. |

### 3.5.3 Attribute Mode Control (Index 0x10)

| Bits | Function |
|---|---|
| 7 | **Not Implemented** (P5:4 Select) |
| 6 | **Not Implemented** (Pel Width) |
| 5 | **Pel Panning Compatibility** — When this bit is a 1, the scanlines following a line compare are immune to the effects of the Pel Panning register. When this bit is a 0, the entire screen is affected by Pel Panning, regardless of the line compare operation. |
| 4 | **Reserved** |
| 3 | **Enable Blink** — When this bit is a 1, attribute bit 7 is used to toggle between the upper and lower halves of the EGA palette every 16 screens, producing a blinking effect. When this bit is 0, attribute bit 7 is used as a background intensity bit. |
| 2 | **Enable Line Graphics Codes** — When this bit is 0, the 9th dot in 9-wide character modes is always set to the background color. When this bit is 1, the 9th dot will be equal to the foreground color for character codes 0xC0-0xDF which are the line graphics character codes. |
| 1 | **Reserved**. |
| 0 | **Graphics Mode** — When this bit is 1, graphics mode is selected and pixel data from the frame buffer is used to produce the pixel stream. When this bit is 0, text mode is selected, and text attribute and font pattern information is used to produce the pixel stream. |

### 3.5.4 Overscan Color (Index 0x11)

| Bits | Function |
|---|---|
| 7:0 | **Overscan Color** — This value is output as the pixel value to the video DAC when the Display Enable signal from the CRTC is inactive. |

### 3.5.5 Color Plane Enable (Index 0x12)

| Bits | Function |
|---|---|
| 7:4 | **Reserved** |
| 3 | **Enable Color Plane 3** — Enables color plane 3. This bit is ANDed with its corresponding pixel bit and the resulting 4-bit value is used as the address into the EGA palette. |
| 2 | **Enable Color Plane 2** — Enables color plane 2. This bit is ANDed with its corresponding pixel bit and the resulting 4-bit value is used as the address into the EGA palette. |
| 1 | **Enable Color Plane 1** — Enables color plane 1. This bit is ANDed with its corresponding pixel bit and the resulting 4-bit value is used as the address into the EGA palette. |
| 0 | **Enable Color Plane 0** — Enables color plane 0. This bit is ANDed with its corresponding pixel bit and the resulting 4-bit value is used as the address into the EGA palette. |

### 3.5.6 Horizontal Pel Panning (Index 0x13)

| Bits | Function |
|---|---|
| 7:4 | **Reserved** |
| 3:0 | **Horizontal Pel Panning** — This field specifies how many pixels the screen image should be shifted to the left by. The table below shows the encoding of the bits. |

| Register Value | Mode 0x13 Panning | 9-wide Text Mode Panning | Panning for All Other Modes |
|---|---|---|---|
| 0 | 0 | 1 | 0 |
| 1 | - | 2 | 1 |
| 2 | 1 | 3 | 2 |
| 3 | - | 4 | 3 |
| 4 | 2 | 5 | 4 |
| 5 | - | 6 | 5 |
| 6 | 3 | 7 | 6 |
| 7 | - | 8 | 7 |
| 8 | - | 0 | - |

### 3.5.7 Color Select (Index 0x14)

*P7 - P6 (bits 3 - 2)*

These bits are used to provide the upper 2 bits of the 8-bit pixel value sent to the video DAC in all modes except the 256 color mode (mode 0x13).

*P5 - P4 (bits 1 - 0)*

These bits are used to provide bits 5 and 4 of the 8-bit pixel value sent to the video DAC when the P4:5 Select bit is set in the Attribute Mode Control Register. In this case they replace bits 5 & 4 coming from the EGA palette.

| Bits | Function |
|---|---|
| 7:4 | **Reserved** |
| 3 | **P7** |
| 2 | **P6** |
| 1 | **P5** |
| 0 | **P4** |

## 3.6 Video DAC Registers

Video DAC palette registers are accessed by writing the Palette Address register at the read or write address, then performing three reads or writes, one for each of the red, green, and blue color values. The video DAC provides an address increment feature that allows multiple sets of color triplets to be read or written without writing the palette address register again. To invoke this feature, simply follow the first triplet read/write with the next triplet read/write.

The original IBM video DAC behavior for read operations is:

1. CPU initiates a palette read by writing INDEX to I/O address 0x3C7
2. Video DAC loads a temporary register with the value stored at palette[INDEX]
3. CPU reads red, green, blue color values from temporary register at I/O address 0x3C9
4. Video DAC increments INDEX (INDEX=INDEX+1)
5. Loop to step 2

The original IBM video DAC behavior for write operations is:

1. CPU initiates a palette write by writing INDEX to I/O address 0x3C8
2. CPU writes red, green, blue color values to temporary DAC registers at I/O address 0x3C9
3. Video DAC stores the temporary register contents in palette[INDEX]
4. Video DAC increments INDEX (INDEX=INDEX+1)
5. Loop to step 2

**Table 3-9 Video DAC Registers Overview**

| Register | R/W | I/O Addr |
|---|---|---|
| Palette Address (Write Mode) | R/W | 0x3C8 |
| Palette Address (Read Mode) | W | 0x3C7 |
| DAC State | R | 0x3C7 |
| Palette Data | R/W | 0x3C9 |
| Pel Mask | R/? | 0x3C6 |

### 3.6.1 Palette Address Register

*Read Address:* 0x3C8
*Write Address:* 0x3C8 (Palette Write Mode), 0x3C7 (Palette Read Mode)

| Bits | Function |
|---|---|
| 7:0 | **Palette Address** |

### 3.6.2 DAC State Register

*Read Address:* 0x3C7

| Bits | Function |
|---|---|
| 7:2 | **Reserved** |
| 1:0 | **DAC State** — This register returns the DAC state for save/restore operations. If the last DAC operation was a palette location read, both bits are 1 (value=11). If the last operation was a palette location write, both bits are 0 (value=00). |

### 3.6.3 Palette Data Register

*Read Address:* 0x3C9
*Write Address:* 0x3C9

| Bits | Function |
|---|---|
| 7:6 | **Reserved** |
| 5:0 | **Color Component Value** — This is a 6-bit color component value that will drive the video DAC for the appropriate color component when the current palette write address is used to address the video DAC in the pixel stream. |

### 3.6.4 Pel Mask Register

*Read Address:* 0x3C6
*Write Address:* 0x3C6

| Bits | Function |
|---|---|
| 7:0 | **Pel Mask** — These bits are used to mask off any of the 8 incoming pixel bits before they are used to address the DAC palette. A 1 enables a pixel bit, a 0 disables it. The mask is ANDed with the pixel value and the result used as the address for the palette. The original VGA DAC had a design flaw that sometimes caused palette data corruption when this register was written. As a result, IBM recommended that this register never be written by applications; however, the Cyrix implementation supports full read/write functionality in this register. |

# Video DAC Registers

# 4 Extended Registers

This chapter documents the set of extended VGA registers that have been added to support functionality beyond that provided by the base VGA. These registers perform the following purpose:

- Return status and system configuration information
- Support VESA BIOS modes
- Support various DAC and flat-panel configurations

All extended registers reside in the CRT Controller register space, and occupy index ranges beyond those defined by the VGA. The table below lists all the extended registers and their CRTC indexes.

**Table 4-1 Extended Registers Summary**

| Register | CRTC Index | Comments |
|---|---|---|
| ExtendedRegisterLock | 0x30 | Unlock code is 0x57 followed by 0x4C |
| SoftVGARevision | 0x31 | Version number for SoftVGA (RO) |
| DisplayDataChannel | 0x32 | Used to read EDID block |
| DirectDrawMemoryBase | 0x3C | Base of off-screen memory in 16K units (RO) |
| DirectDrawMemorySize | 0x3D | Size of off-screen memory in 16K units (RO) |
| GraphicsMemorySize | 0x3E | Size of graphics memory in 64K units (RO) |
| ModeSwitchControl | 0x3F | Used by the BIOS to “bookend” mode switches |
| ModeNumber | 0x40 | Used by BIOS to store mode number |
| VerticalTimingExtension | 0x41 | |
| HorizontalTimingExtension | 0x42 | |
| ExtendedAddressControl | 0x43 | |
| ExtendedStartAddress | 0x44 | |
| ExtendedOffset | 0x45 | |
| ExtendedColorControl | 0x46 | |
| WriteMemoryAperture | 0x47 | |
| ReadMemoryAperture | 0x48 | |
| DisplayCompression | 0x49 | |
| DriverControl | 0x4A | |
| DacControl | 0x4B | |
| ClockControl | 0x4C | |
| CrtClockFrequency | 0x4D | |
| CrtClockFrequencyFraction | 0x4E | |
| RefreshRate | 0x4F | |
| DisplayEnable | 0x50 | |

**Table 4-1 Extended Registers Summary (cont.)**

| Register | CRTC Index | Comments |
|---|---|---|
| FpType | 0x51 | |
| FpResolution | 0x52 | |
| FpControl | 0x53 | |
| FpDither | 0x54 | |
| FpClockFrequency | 0x55 | |
| FpHorizontalTotal | 0x56 | |
| FpVerticalTotal | 0x57 | |
| FpHorizontalSyncStart | 0x58 | |
| FpHoprizontalSyncEnd | 0x59 | |
| FpVerticalSyncStart | 0x5A | |
| FpVerticalSyncEnd | 0x5B | |
| CrtHorizontalSyncStart | 0x5C | |
| CrtHorizontalSyncEnd | 0x5D | |
| CrtVerticalSyncStart | 0x5E | |
| CrtVerticalSyncEnd | 0x5F | |
| PanOffsetLow | 0x60 | |
| PanOffsetHigh | 0x61 | |
| PanOffsetExtended | 0x62 | |
| ExtendedFontControl | 0x63 | |
| TVControl | 0x67 | |
| TVMode | 0x68 | |
| TVHorzPosition | 0x69 | |
| TVVertPosition | 0x6A | |
| TVBrightness | 0x6B | |
| TVEncoder | 0x6C | |

Since the registers in the range 0x40-0x4F are mode specific they are saved/restored by the mini-VDD during screen switching under Windows 95 (all other registers are typically system-related and are not saved/restored). This should also apply to other operating systems with a similar methodology for switching between different modes.

The DisplayEnable register provides an example of why the system registers are not saved/restored. If the user hits a hot key to switch from a CRT only display to a simultaneous display, the DisplayEnable register is modified to reflect this. If this register were restored on a screen switch, then the system would revert to the CRT-only display.

## 4.1 ExtendedRegisterLock (Index 0x30)

| Bits | Function |
|---|---|
| 7:0 | **Lock** — Writing 0x57 followed by 0x4C will unlock the extended registers. Writing any other value will lock them. |

This register controls access to all extended VGA registers. At power-up, extended registers are inaccessible (except for this one). To unlock them, the user must write two consecutive bytes to this register: 0x57 followed by 0x4C. The extended registers will then be available for reading and writing. To relock the extended registers, the user writes any other value to this register.

If the registers are currently locked, a read to this register will return 0xFF. If they are unlocked, a read will return 0.

## 4.2 SoftVgaRevision (Index 0x31)

| Bits | Function |
|---|---|
| 7:0 | **Revision Number** — This field returns the revision number for SoftVGA. Released products start at version 1; a value of 0 indicates a prerelease or non-standard version. |

## 4.3 DisplayDataChannel (Index 0x32)

| Bits | Function |
|---|---|
| 7:0 | **Extended Display Interface Data (EDID)** — On writes, the value specifies which of 8 structures to read (indexed 0 -7) and resets the address pointer to the first byte in the structure. On reads, the current byte within the EDID structure is returned and the address pointer is incremented to the next byte. |

## 4.4 DirectDrawMemoryBase (Index 0x3C)

| Bits | Function |
|---|---|
| 7:0 | **Off Screen Memory Base** — The value read from this register is an offset from the beginning of graphics memory in units of 16KB. This memory is intended to be used by the DirectDraw driver, and is not intended to be used by the video BIOS to provide off-screen memory to VESA applications. |

## 4.5 DirectDrawMemorySize (Index 0x3D)

| Bits | Function |
|---|---|
| 7:0 | **Off Screen Memory Size** — The value read from this register is the size of off-screen memory available to the DirectDraw driver in units of 16KB. This is not intended to be used by the video BIOS to provide off-screen memory to VESA applications. |

## 4.6 GraphicsMemorySize (Index 0x3E)

| Bits | Function |
|---|---|
| 7:0 | **Graphics Memory Size** — The value read from this register is the total amount of graphics memory available in 64KB increments. |

## 4.7 ModeSwitchControl (Index 0x3F)

| Bits | Function |
|---|---|
| 7:1 | **Reserved** |
| 0 | **Mode Switch Active** — This bit is set by video BIOS at the beginning of a mode switch, and cleared at the end of the mode switch. VGA does not validate the new register state until this bit is 0. |

## 4.8 ModeNumber (Index 0x40)

| Bits | Function |
|---|---|
| 7:0 | **Mode Number** — This is a scratch register provided for use by the video BIOS to store the current mode number. It is not used by VGA. |

## 4.9 VerticalTimingExtension (Index 0x41)

| Bits | Function |
|---|---|
| 7 | **Reserved** |
| 6 | **Vertical Sync Start bit 10** — This bit extends the VGA Vertical Sync Start value to 11 bits. Bits 7:0 are in Vertical Sync Start, and bits 9:8 are in Overflow. |
| 5 | **Reserved** |
| 4 | **Vertical Blank Start bit 10** — This bit extends the VGA Vertical Blank Start value to 11 bits. Bits 7:0 are in Vertical Blank Start, bit 8 is in Overflow, bit 9 is in Maximum Scan Line. |
| 3 | **Reserved** |
| 2 | **Vertical Display End bit 10** — This bit extends the Vertical Display End value to 11 bits. Bits 7:0 are in Vertical Display End, bits 9:8 are in Overflow. |
| 1 | **Reserved** |
| 0 | **Vertical Total bit 10** — This bit extends the Vertical Total value to 11 bits. Bits 7:0 are in Vertical Total, bits 9:8 are in Overflow.\ |

## 4.10 HorizontalTimingExtension (Index 0x42)

This register is currently not implemented, and is reserved for future enhancements.

## 4.11 ExtendedAddressControl (Index 0x43)

| Bits | Function |
|---|---|
| 7:3 | **Reserved** |
| 2 | **Pixel Double** — Indicates that pixel doubling is enabled. This is used for low resolution modes. |
| 1 | **Direct Frame Buffer** — If this bit is 1, the VGA front end draws directly into the displayed frame buffer. Otherwise, the front end draws into a separate VGA frame buffer, and that frame buffer is then translated into the displayed frame buffer. |
| 0 | **Packed Chain4** — This bit puts the VGA front end into true packed-pixel mode. Consecutive bytes in the host address space map to consecutive bytes in the frame buffer (unlike the standard mode 13h mapping). This bit takes precedence over Chain4, Chain Odd/Even, and both Odd/Even bits. |

This register contains extensions for VGA addressing control. These extensions are intended for use by VESA BIOS code.

The packed-pixel organization of the frame buffer in a VESA BIOS mode is not supported by any standard VGA configurations. VGA Chain4 mode places entire bytes directly into the frame buffer, but uses only one out of every four doublewords. The packed Chain4 bit enables a Chain4 mode that doesn't skip doublewords, allowing the VGA write pipe to send pixels directly to a VESA-formatted frame buffer. (Note: some VGA chips, including the Compaq QVision and Tseng W32P do this as their native Chain4 modes).

Once Packed Chain4 is enabled, there is no longer a need for a separate VGA frame buffer, since the pixel format coming from the VGA write pipe is appropriate for the displayed frame buffer. The DirectFrame Buffer bit has two effects. The first is that the VGA write pipe will write directly into the displayed frame buffer (see the Memory Aperture register for more details). The second is that frame buffer writes no longer cause SMIs, since SoftVGA no longer needs to translate pixel data.

When Direct Frame Buffer is set, certain parts of the standard VGA register set have no effect. In particular, the attribute controller registers are ignored, including the 16-entry color palette. All panning operations (except for changing the start address) are disabled, as is Line Compare (split screen). The Word/Byte and Doubleword bits in the CRTC are ignored; the VGA is effectively in Byte mode.

## 4.12 ExtendedStartAddress (Index 0x44)

| Bits | Function |
|---|---|
| 7 | **Lock** — This bit is used to lock the current start address while loading a new value. This prevents the hardware from using intermediate values that are generated when writing the three registers required to set a new start address (0xC, 0xD, 0x44). |
| 6:5 | **0** |
| 4:0 | **Start Address 20:16** — This field extends the CRTC Start Address to 21 bits. bits 15:8 are in Start Address High, and bits 7:0 are in Start Address Low. |

**Table 4-2** **ExtendedOffset (Index 0x45)**

| Bits | Function |
|---|---|
| 7:2 | **Reserved** |
| 1:0 | **Offset bit 9:8** — This field extends the CRTC offset to 10 bits. |

## 4.13 ExtendedColorControl (Index 0x46)

This register contains fields for color control in extended modes. In 16 BPP modes, each pixel in the displayed frame buffer occupies two bytes. No palette or color map applies. Instead, the 16-bit quantity contains the red, green, and blue color values as bit fields. These bit fields can be arranged in two ways: 555 or 565 as in Table 4-3.

**Table 4-3 555 / 565 Format**

| Bits | Color |
|---|---|
| **555 Format** | |
| 15 | X |
| 14:10 | Red |
| 9:5 | Green |
| 4:0 | Blue |
| **565 Format** | |
| 15:11 | Red |
| 10:5 | Green |
| 4:0 | Blue |

Standard VGA modes all expect the displayed frame buffer to be 8 BPP. The 16BPP bit must be cleared for them to operate properly. Only VESA BIOS should set bits in this register.

**Table 4-4 GXm ExtendedColor Register Bits**

| Bits | Function |
|---|---|
| 7:2 | **Reserved** |
| 1 | **555 Format** — In 16BPP mode, pixel format is 555 (bit=1) rather than 565 (bit=0). |
| 0 | **16 Bits Per Pixel** — When 1, each displayed frame buffer pixel is formed from two consecutive bytes (16 bits per pixel). |

**Table 4-5 MXi ExtendedColor Register Bits**

| Bits | Function |
|---|---|
| 7:3 | **Reserved** |
| 2 | **32 Bits Per Pixel** — Indicates 32BPP. |
| 1 | **555 Format** — In 16BPP mode, pixel format is 555 (bit=1) rather than 565 (bit=0). |
| 0 | **16 Bits Per Pixel** — When 1, each displayed frame buffer pixel is formed from two consecutive bytes (16 bits per pixel). |

## 4.14 WriteMemoryAperture (Index 0x47)

| Bits | Function |
|---|---|
| 7:5 | **Reserved** |
| 4:0 | **Write Aperture Base** — If DirectFrameBuffer is set, this field specifies the mapping of the 64KB aperture at A0000h into the displayed frame buffer. This offset is specified in 64KB units. |

This register controls the address base for frame buffer accesses in DirectFrameBuffer mode. (See the ExtendedAddressControl register.) Writes to the VGA frame buffer address range (normally A0000h-AFFFFh for VESA modes) will be sent to the 64KB aperture beginning at offset (WriteApertureBase * 0x10000) bytes from the base of the displayed frame buffer.

## 4.15 ReadMemoryAperture (Index 0x48)

| Bits | Function |
|---|---|
| 7:5 | **Reserved** |
| 4:0 | **Read Aperture Base** — If DirectFrameBuffer is set, this field specifies the mapping of the 64KB aperture at A0000h into the displayed frame buffer. This offset is specified in 64KB units. |

This register controls the address base for frame buffer accesses in DirectFrameBuffer mode. (See the ExtendedAddressControl register.) Reads to the VGA frame buffer address range (normally A0000h-AFFFFh for VESA modes) will be sent to the 64KB aperture beginning at offset (ReadApertureBase * 0x10000) bytes from the base of the displayed frame buffer.

## 4.16 DisplayCompression (Index 0x49)

| Bits | Function |
|---|---|
| 7:4 | **Reserved** |
| 3:1 | **Static Frame Select** — This field specifies the number of static frames to insert between update frames. A static frame is referred to as a frame in which dirty lines will not be updated. |
| 0 | **Display Compression Enable** — This bit enables the hardware display compression. Compression is only enabled if the current graphics memory organization has allocated memory for the compression buffer. |

## 4.17 DriverControl (Index 0x4A)

| Bits | Function |
|---|---|
| 7:1 | **Reserved** |
| 0 | **Display Driver Active** — This bit indicates to the VGA that a display driver is controlling the cursor, so it (the VGA) should not set the cursor start offset of other cursor parameters that may interfere with the driver. When set, SoftVGA will not set the cursor or palette registers. This allows the display driver to have direct control over the hardware cursor and palette. |

## 4.18 DACControl (Index 0x4B)

| Bits | Function |
|---|---|
| 7:4 | **Reserved** |
| 3 | **1/2 Pixel Per Clock** — When this bit is set, a 16-bit pixel is sent to the 8-bit RAMDAC port in two 8-bit pieces. The RAMDAC clock is driven at twice the speed of the dot clock. |
| 2 | **2 Pixels Per Clock** — When this bit is set, two 8-bit pixels are sent to the 16-bit RAMDAC port every clock. The RAMDAC clock is driven at half the speed of the dot clock. |
| 1 | **Enable 16-bit Bus** — This bit indicates that the current video mode requires a 16-bit bus ????. |
| 0 | **Reserved** |

## 4.19 ClockControl (Index 0x4C)

| Bits | Function |
|---|---|
| 7 | **Extended Clock Mode** — When this bit is set, the ClockFrequency register is used to program the PLL that generates the dot clock. Otherwise, the dot clock is determined by the Miscellaneous Output Register Clock Select field. |
| 6 | **Reserved** |
| 5 | **Halve Dot Clock** — When this bit is set, the external dot clock is divided by two to provide the internal dot clock. |
| 4 | **Double Dot Clock** — When this bit is set, the external dot clock is doubled to provide the internal dot clock. |
| 3 | **Reserved** |
| 2:0 | **GENDAC Type** — This field indicates the type of clock generator/RAMDAC being used. The only currently used value is 0=ICS5342. Values 1-7 are reserved. |

## 4.20 CrtClockFrequency (Index 0x4D)

| Bits | Function |
|---|---|
| 7:0 | **Clock Frequency** — This field indicates the integer portion of the clock frequency in MHz used for the dot clock. It is combined with register 0x4E to create a fixed point value (8.8). This clock frequency is used when the flat panel is disabled. |

## 4.21 CrtClockFrequencyFraction (Index 0x4E)

| Bits | Function |
|---|---|
| 7:0 | **Clock Frequency Fraction** — This field contains the fractional portion of the clock frequency in MHz used for the dot clock. It is combined with register 0x4D to create a fixed point value (8.8). This clock frequency is used when the flat panel is disabled. |

## 4.22 RefreshRate (Index 0x4F)

| Bits | Function |
|---|---|
| 7:0 | **Refresh Rate** — The desired refresh rate, as specified by the display driver. |

This register does not actively set a particular refresh rate.

Before setting a display mode, the display driver uses an OEM-specific call to the video BIOS to select the desired refresh rate. For that function call, it simply writes the value into this register. It then reads the register when setting the video timings to select the appropriate table to use. The following VESA call selects the refresh rate.

```
mov ax, 4F14h        ; OEM extension
mov bx, 0000h        ; refresh rate select
mov dx, rate
int 10h
```

## 4.23 DisplayEnable (Index 0x50)

| Bits | Function |
|---|---|
| 7:6 | **Reserved** |
| 5 | **Disable CRT VSYNC** — This bit disables vertical sync to the CRT. The video BIOS sets this bit for DPMS. |
| 4 | **Disable CRT HSYNC** — This bit disables horizontal sync to the CRT. The video BIOS sets this bit for DPMS. |
| 3 | **Reserved** |
| 2 | **Fixed Timing Enable** — If set, the fixed timings are used for all display modes. These timing are specified by in the Extended Registers 53h-5Fh. |
| 1 | **CRT Enable** — Enable the CRT for display. |
| 0 | **Flat Panel Enable** — Enable the Flat Panel for display. |

The flat panel timings are defined using the extended CRTC registers (0x53-0x5F). These timings are used when the flat panel is enabled (bit 0 set). The flat panel timings are also used for simultaneous display with the CRT (bit 0 and bit 1 set). If the flat panel is disabled by clearing bit 0, the panel is powered down by VGA before setting the CRT timings. If both the CRT and flat panel are disabled (bit 0 and 1 cleared), then the entire video subsystem is shut down to conserve power.

## 4.24 FpResolution (Index 0x52)

| Bits | Function |
|---|---|
| 7:2 | **Reserved** |
| 1:0 | **Resolution** — This field indicates the resolution of the flat panel as follows:<br>00 = 640x480<br>01 = 800x600<br>10 = 1024x768<br>11 = Reserved |

The system BIOS sets this register during POST to indicate the resolution of the flat panel. This information is then used to generate the appropriate timings.

## 4.25 FpControl (Index 0x53)

| Bits | Function |
|---|---|
| 7:6 | **Reserved** |
| 5 | **FP Vertical Sync Polarity** — Set for positive vertical sync. |
| 4 | **FP Horizontal Sync Polarity** — Set for positive horizontal sync. |
| 3 | **Reserved** |
| 2:0 | **Power Sequence Delay** — This field sets the delay between edges of the power sequencing delay logic. The value is multiplied by one display frame period. |

## 4.26 FpDither (Index 0x54)

| Bits | Function |
|---|---|
| 7:4 | **Reserved** |
| 3 | **3/4 Bit Color** — When this bit is set, 4 bits are used for dithering, along with frame rate modulation. When this bit is clear, 3 bits are used for dithering. |
| 2 | **Dither Enable** — When this bit is set, a 2x2 spatial dither pattern is enabled. |
| 1 | **Frame Rate Modulation Select** — When this bit is set, the pattern changes every two display frames rather than every frame. |
| 0 | **Frame Rate Modulation Enable** — When this bit is set, frame rate modulation is enabled. |

## 4.27 FpClockFrequency (Index 0x55)

| Bits | Function |
|---|---|
| 7:0 | **FP Clock Frequency** — This field contains the integer number of MHz for the dotclock when the flat panel is enabled. Note that there is no fractional component for this number. |

## 4.28 FpHorizontalTotal (Index 0x56)

| Bits | Function |
|---|---|
| 7:0 | **FP Horizontal Total** — This field specifies the number of characters (8 pixels each) between the end of the active display and the end of the total horizontal period. |

This register is used to generate timings for the flat panel.

## 4.29 FpVerticalTotal (Index 0x57)

| Bits | Function |
|---|---|
| 7:0 | **FP Vertical Total** — This field specifies the number of lines between the end of the active display and the end of the total vertical period. |

This register is used to generate timings for the flat panel.

## 4.30 FpHorizontalSyncStart (Index 0x58)

| Bits | Function |
|---|---|
| 7:0 | **FP Horizontal Sync Start** — This field specifies the number of characters (8 pixels each) between the end of the active display and the start of the horizontal sync pulse. |

This register is used to generate timings for the flat panel.

## 4.31 FpHorizontalSyncEnd (Index 0x59)

| Bits | Function |
|---|---|
| 7:0 | **FP Horizontal Sync End** — This field specifies the number of characters (8 pixels each) between the end of the active display and the end of the horizontal sync pulse. |

This register is used to generate timings for the flat panel.

## 4.32 FpVerticalSyncStart (Index 0x5A)

| Bits | Function |
|---|---|
| 7:0 | **FP Vertical Sync Start** — This field specifies the number of lines between the end of the active display and the start of the vertical sync pulse. |

This register is used to generate timings for the flat panel.

## 4.33 FpVerticalSyncEnd (Index 0x5B)

| Bits | Function |
|---|---|
| 7:0 | **FP Vertical Sync End** — This field specifies the number of lines between the end of the active display and the end of the vertical sync pulse. |

This register is used to generate timings for the flat panel.

## 4.34 CrtHorizontalSyncStart (Index 0x5C)

| Bits | Function |
|---|---|
| 7:0 | **CRT Horizontal Sync Start** — This field specifies the number of characters (8 pixels each) between the end of the active display and the start of the horizontal sync pulse. |

This register is used to generate timings for the CRT when both the CRT and flat panel are enabled.

## 4.35 CrtHorizontalSyncEnd (Index 0x5D)

| Bits | Function |
|---|---|
| 7:0 | **CRT Horizontal Sync End** — This field specifies the number of characters (8 pixels each) between the end of the active display and the end of the horizontal sync pulse. |

This register is used to generate timings for the CRT when both the CRT and flat panel are enabled.

## Table 4-6 CrtVerticalSyncStart (Index 0x5E)

| Bits | Function |
|---|---|
| 7:0 | **CRT Vertical Sync Start** — This field specifies the number of lines between the end of the active display and the start of the vertical sync pulse. |

This register is used to generate timings for the CRT when both the CRT and flat panel are enabled

## 4.36 CrtVerticalSyncEnd (Index 0x5F)

| Bits | Function |
|---|---|
| 7:0 | **CRT Vertical Sync End** — This field specifies the number of lines between the end of the active display and the end of the vertical sync pulse. |

This register is used to generate timings for the CRT when both the CRT and flat panel are enabled.

## 4.37 PanOffsetLow (Index 0x60)

| Bits | Function |
|---|---|
| 7:0 | **Pan Offset bits 7:0** — This is the low byte of the Pan Offset. |

## 4.38 PanOffsetHigh (Index 0x61)

| Bits | Function |
|---|---|
| 7:0 | **Pan Offset bits 15:8** — This is the middle byte of the Pan Offset. |

## 4.39 PanOffsetExtended (Index 0x62)

| Bits | Function |
|---|---|
| 7:0 | **Pan Offset bits 23:16** — This is the high-order byte of the Pan Offset. |

The Pan Offset is added to the current display start offset when a flat panel is enabled. This allows resolutions that are greater than the resolution of the flat panel to be displayed entirely for a CRT-only display and panned when the flat panel is enabled.

## 4.40 ExtendedFontControl (Index 0x63)

| Bits | Function |
|---|---|
| 7 | **8-Dot Character Select** — When 1, the normal 8-dot/9-dot character select value will be forced to display 8-dot characters. This allows standard VGA text mode 3 to be displayed on a 640x480 flat panel. |
| 6:3 | **Reserved** |
| 2:0 | **Vertical Expansion** — This field specifies the number of lines to vertically expand the character size. The last line is duplicated for graphics characters and blank lines are inserted for regular characters. |

This register is used to modify the appearance of text modes on flat panels. The video BIOS sets the 8-dot Character Select bit when running on a 640x480 flat panel. This bit is used rather than setting the 8/9 Dot Clock bit in the Clocking Mode Register (SR1) to maintain VGA compatibility. Some VGA fields have different meanings in 8-dot or 9-dot character modes.

For GX, the Vertical Character Expansion field is set by the video BIOS when running on a flat panel since hardware stretching is not an option. In 400 scanline text modes the characters are expanded by three lines to generate a display of 475 scanlines. This almost fills the screen on a 640x480 panel.

The Vertical Character Expansion field is ignored, however, if an application sets the mode to 8-dot characters using the Clocking Mode Register (SR1). Applications that do this may be using special fonts to display graphics in a text mode (startup screens, for example). In that case, the extra lines would split the graphics image and ruin the intended visual effect.

## 4.41 TVControl (Index 0x67)

| Bits | Function |
|---|---|
| 7 | **NTSC/PAL** — This field specifies PAL timings (bit=1) or NTSC timings (bit=0). The system BIOS uses CMOS settings to set the appropriate value of this bit before initializing the video BIOS. This bit is also used by the BIOS to program the appropriate display mode in the video encoder chip. |
| 6 | **Force CRT** — If this bit is set, the video BIOS will skip the TV detection sequence. For TV encoder chips that do not have support for TV detection, the video BIOS will always return that a TV is attached unless this bit is set. |
| 5 | **Reserved** |
| 4 | **Contrast Enable** — This bit enables the contrast control circuitry on the TV encoder. It is set by the BIOS as boot. Applications can use a video BIOS call to change the value of this bit, but should not change it directly (the video BIOS knows to reprogram the encoder chip as well). |
| 3 | **Composite Enable** — This bit enables the composite video output. It is set by the system BIOS at boot based on the detection results. Applications can use a video BIOS call to change the value of this bit, but should not change it directly. |
| 2 | **S-Video Enable** — This bit enables S-Video output. It is set by the system BIOS at boot based on the detection results. Applications can use a video BIOS call to change the value of this bit, but should not change it directly. |
| 1 | **TV Timings** — This bit indicates that SoftVGA should use the fixed timings specified in registers 54h-5Fh. It is set by the system BIOS at boot based on the detection results. |
| 0 | **TV Attached** — This bit indicates that a TV was detected during the boot sequence. It is set by the system BIOS |

The system BIOS sets bits 6 & 7 before initializing the video BIOS. If bit 6 is set, the video BIOS does not perform TV detection, and bits 0-3 are cleared. If bit 6 is clear, the video BIOS performs TV detection and sets bits 0-3 appropriately. Bit 7 is used by the video BIOS when loading the fixed timing values (regs 54h-5Fh).

## 4.42 TVMode (Index 0x68)

| Bits | Function |
|---|---|
| 7:2 | **Reserved** |
| 1 | **TV Resolution** — This bit determines the TV screen resolution 1=800x600, 0=640x480. This bit is cleared by the video BIOS at boot. It can be changed by an application making a call to the video BIOS. This BIOS uses this bit to pick the appropriate timing values and program the TV encoder properly. |
| 0 | **Underscan Select** — This bit determines underscan mode (bit=1) or overscan mode (bit=0). This bit is cleared by the video BIOS at boot. It can be changed by an application making a call to the video BIOS. The BIOS uses this bit to pick the appropriate timing values and program the TV encoder properly |

## 4.43 TVHorizontalPosition (Index 0x69)

| Bits | Function |
|---|---|
| 7:0 | **Horizontal Position** — This field is a signed value used to shift the TV image left (minus values) or right (plus values). |

## 4.44 TVVerticalPosition (Index 0x6A)

| Bits | Function |
|---|---|
| 7:0 | **Vertical Position** — This field is a signed value that shifts the TV image up (negative numbers) or down (positive numbers). |

## 4.45 TVBrightness (Index 0x6B)

| Bits | Function |
|---|---|
| 7:0 | **Brightness** — This field specifies the TV brightness level (0 is low, 255 is high). |

## 4.46 TVEncoder (Index 0x6C)

| Bits | Function |
|---|---|
| 7:2 | Reserved |
| 1 | Brightness Control — Encoder supports brightness control |
| 0 | Position Control — Encoder supports position control |

This register specifies the functionality of the attached encoder. The system BIOS sets this at POST based on the type of TV encoder in the system.

A value of 0 implies that no TV encoder is present in the system.