RGB640

170/220 MHz
High Performance
Palette DAC

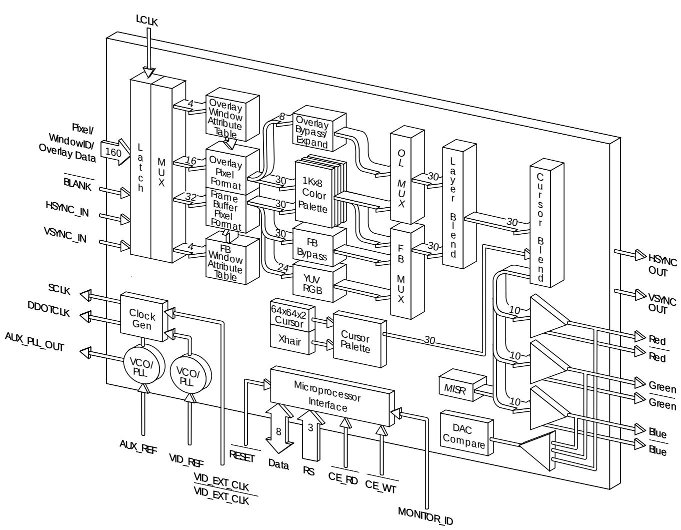
Product Description

The RGB640™ High Performance Palette DAC from IBM combines a comprehensive array of high-end display subsystem features with real-time layer-blending and color space conversion circuitry for the concurrent display of 3D animation, true-color digital video, business graphics and photorealistic natural images.

The RGB640 provides individual pixel formats for up to 16 overlay windows. Start-skip configuration of the 160-bit memory interface maximizes frame buffer design flexibility. Double-buffered primary and overlay layers enhance 3D applicaitons. Mixed YUV and RGB formats and fully-programmable YUV-RGB conversion permit simultaneous display of graphics and video pixel streams. Programmable on-chip blending of the primary and overlay layers enables active use of multiple windows.

The RGB640's 10-bit DAC and dual PLL clock synthesizers provide vibrant color with unsurpassed image stability and bright, flicker-free display even on large, high-resolution monitors.

Functional Block Diagram



Product Highlights

- ❑ 170, 220 MHz operation
- ❑ Triple monotonic 10-bit DACs
- ❑ Fully programmable YUV-RGB conversion
- ❑ Mixed YUV and RGB pixels
- ❑ 160-bit wide pixel data bus
- ❑ 2:1, 4:1, 5:1, 8:1, 16:1, 16:3 multiplexing
- ❑ Display modes up to 1600x1280
- ❑ Large Screen ISO-compliant refresh rates
- ❑ 4/8/10/16/24/30-bits per pixel
- ❑ Gamma correction
- ❑ 1024-shade gray scale
- ❑ Three 1K x 10 color palette RAMs
- ❑ Two on-chip clock generators
- ❑ Up to 16 user-defined primary and overlay window types
- ❑ 64x64/32x32 translucent hardware cursor
- ❑ On-chip crosshair
- ❑ Double-buffered primary and overlay layers
- ❑ Primary/overlay layer blending
- ❑ Chroma-key
- ❑ Pixel interleave
- ❑ 8-bit VGA Data Input
- ❑ On-chip diagnostic functions
- ❑ Low-power 3.3V operation
- ❑ 5V-tolerant inputs
- ❑ 272-pin QFP / 304-pin BGA package
- ❑ 0.8 μm CMOS

Applications

- ❑ Graphical user interfaces
- ❑ Video playback and post-processing
- ❑ 3D CAD/CAM, games and virtual reality
- ❑ Imaging
- ❑ Scientific Visualization

Table of Contents

List of Figures

List of Tables

1.0 Introduction

The RGB640 is a high performance palette DAC which accepts digital pixels and converts the data to analog form. A block diagram is given in Figure 1 on page 3.

The RGB640 combines features used in advanced workstations and high end PCs.

1.1 Workstation Features

1.1.1 10 Bit RGB DACs

Internally, the RGB640 supports red, green and blue digital pixels of 10 bits each. Each of the three on-board DACs (Digital to Analog Converters) converts a 10-bit digital value to analog form. The DACs are monotonic. Composite sync-on-green is supported.

1.1.2 Wide, Flexible Serial Data Input

The incoming serial data bus is 160 inputs, of which 128 inputs are used for pixels and 32 inputs are used for pixels or window ID. In general multiple pixels are latched with a load clock, and then serialized (multiplexed) to form an internal stream of pixels that is clocked at pixel speed.

The serializer is programmable to operate in a variety of modes. For example:

- The multiplex ratio can be 2:1, 4:1, 5:1, 8:1, 16:1 and 16:3. 5:1 mode is typically used by workstations to support a horizontal line width of 1280 pixels. 16:3 mode is used to support "packed" 24-bit pixels.
- Pixel interleave is supported. This changes the serialization order on each horizontal line. This is typically used by the graphics controller to speed up line drawing.

A full description of the serializer is given in section 2.0, "Serializer" on page 5.

1.1.3 Multiple Pixel Formats

A number of pixel formats are supported, including 3/3/2 RGB, 5/5/5 and 5/6/5 RGB, 8/8/8 RGB and 10/10/10 RGB. 4, 8 and 10 bit gray scale formats are supported.

The pixels can be used to index the internal palettes (also known as CLUTs, for Color Look Up Tables). There are three 1K x 10 palettes; one each for red, green, blue.

The pixels can also bypass the palettes. The pixels are expanded in one of several ways, to 10 bits per color.

A description of all the pixel types is given in section 3.0, "Pixel Formats" on page 15.

1.1.4 Window IDs

The RGB640 supports windowed environments in which the various windows displayed on the screen can each have different attributes such as pixel type.

The palette DAC has an internal Window Attribute Table (WAT), which is preloaded with the attributes of the various window types. The frame buffer contains both pixel data and Window ID (WID) data. The WID data is latched by the palette DAC along with the pixel data. The WID is used as an index into the WAT. The attributes read out of the WAT are used to interpret the processing of the pixel data.

The WAT entries also contain a starting address for addressing the palettes. For pixel types which index the palettes, this allows the palettes to be sub-setted, such that various windows can each have "their own" palettes.

The format of the WAT entries is given in section 4.0, "Window Attribute Tables" on page 23.

1.1.5 Double Buffer Operation

Double buffer operation (switching between two images stored in separate buffers) is supported. The incoming pixel data is regarded as coming from two buffers, A and B. Pixels from one buffer are selected and displayed; the data for the other buffer are discarded.

Entries in the WAT table select the A or B buffer (sections 4.1.3 and 4.2.3). A clocking variation for use with double buffer operation called "Load Clock Interleave" is discussed in section 2.5.

1.1.6 Cursor and Crosshair

The RGB640 generates both a cursor and a crosshair.

The cursor is a 32 x 32 or 64 x 64 pattern that is overlaid on the display data. The pattern is held in an onboard 1K x 8 memory.

Each pixel of the cursor can be transparent, highlighted, translucent, or a solid color. The cursor can be made to blink with a programmable rate and duty cycle.

The crosshair is a generated horizontal and vertical line that is overlaid on the display pixels. The lines can be single-colored or patterned. The crosshair can be made to blink, and can be clipped to a window boundary using the WATs.

Full descriptions of the cursor and crosshair are given in section 5.0, "Cursor and Crosshair" on page 29.

1.1.7 Dual Layer Operation

Dual layer operation is supported, in which the incoming pixel data is interpreted as coming from two layers: frame buffer and overlay. Each layer has its own WAT. The two WATs, frame buffer WAT and overlay WAT, can be individually addressed with two separate WIDs, or a single WID can address both WATs.

The frame buffer layer supports all of the operations and pixel types discussed above. The overlay layer is more restricted and does not support all pixel types.

The two layers can be blended together. In this mode of operation one or the other layer appears translucent, such that the other layer "shines through." The blending is adjustable in steps of less than 2% of full range.

Blending is controlled using the overlay WAT (section 4.2.4). If blend is not used, then either the frame buffer layer or the overlay layer can be specified for display.

1.1.8 YUV Format

In addition to the various RGB formats, YUV formats (luminance/chrominance) are supported for the frame buffer layer. For these formats Y is used as a luminance (brightness) value, and U and V are used as chrominance (color difference) values. A programmable color-space converter transforms the YUV pixels to RGB.

The YUV data can be 24-bit (8 bits each for Y, U and V) or 16-bit subsampled (8 bits of Y and 8 bits of alternating U and V). A variation of the 16-bit format uses the high order bit as a tag to interpret the remaining 15 bits as 5/5/5 RGB or 15-bit subsampled YUV.

When dual layer operation is used, a chroma key is available. This allows the selection of the display layer, YUV frame buffer or RGB overlay, by comparing the overlay pixel data to a chroma key register.

The YUV format is discussed in section 3.3, "YUV Formats" on page 15. Chroma key operation is discussed in Section 4.2.4 on page 27.

1.2 PC Features

1.2.1 VGA Port

The RGB640 can be programmed to act as a VGA palette DAC. In this mode of operation 8 of the serial data inputs are used as VGA pixel data. The VGA pixels are masked with a pixel mask register, and used as an index into the palettes.

The RGB640 resets to VGA operation, and must be programmed to switch to full serial data input use. VGA operation is discussed in section 6.0, "VGA" on page 33.

1.2.2 Other

When programmed for non-VGA serial data input, in general, a subset of the features described above provide the formats and operations that are required and can be used by typical PC software. These include the 4, 8, 15/16 and 24-bit pixel formats, both palette indexed and palette bypassed. PC software can typically use the hardware cursor as well.

1.3 Clocking

The RGB640 has two on-board programmable clock generators (programmable Phase Locked Loops). The video PLL is used internally for the high speed pixel clock. SCLK, a divided version of the internal pixel clock, is provided. SCLK is used by external logic to provide LCLK (load clock) to the serializer.

The auxiliary PLL is not used by the RGB640. Its output is provided for general use (e.g., for the graphics display controller clocking).

The programming and operation of the two clock generators is described in section 7.0, "Clocking" on page 35.

1.4 Diagnostics

A MISR (Multiple Input Signature Register) is available for accumulating a cyclical signature of a frame of pixels. Mask registers are available for masking off portions of the serial data inputs. Fault isolation to various components, such as the frame buffer VRAM modules, can be accomplished by masking off selected serial data inputs and comparing the accumulated signatures with known good signatures.

The diagnostic features are discussed in section 8.0, "Diagnostics" on page 39.

Figure 1. RGB640 Data Flow

1.5 Software Interface

In addition to the palette, the WATs, and the cursor map, there are a variety of 8-bit control registers. All of these tables and registers are accessed through an 8-bit processor port.

In general the palette and VGA pixel mask are accessed in a VGA-compatible manner. The WATs, cursor map, and full palette are accessed indirectly using an index register scheme.

Software access in general is discussed in section 9.0, "Software Interface" on page 40. The individual control registers are discussed in section 10.0, "Register Descriptions," starting on page 48.

1.6 Other Features

The features described above are only an overview. See the referenced sections for descriptions of all features provided by the RGB640. Some features are described in the bit descriptions of the control registers discussed in section 10.0. Additional detail is found in section 11.0, "Pin Descriptions" on page 84.

1.7 Bit And Byte Ordering

In this specification, bit 0 is the least significant bit, bit 1 is the next most significant, and so on. Byte 0 is the least significant byte, byte 1 is the next most significant, and so on, as shown in Figure 2.

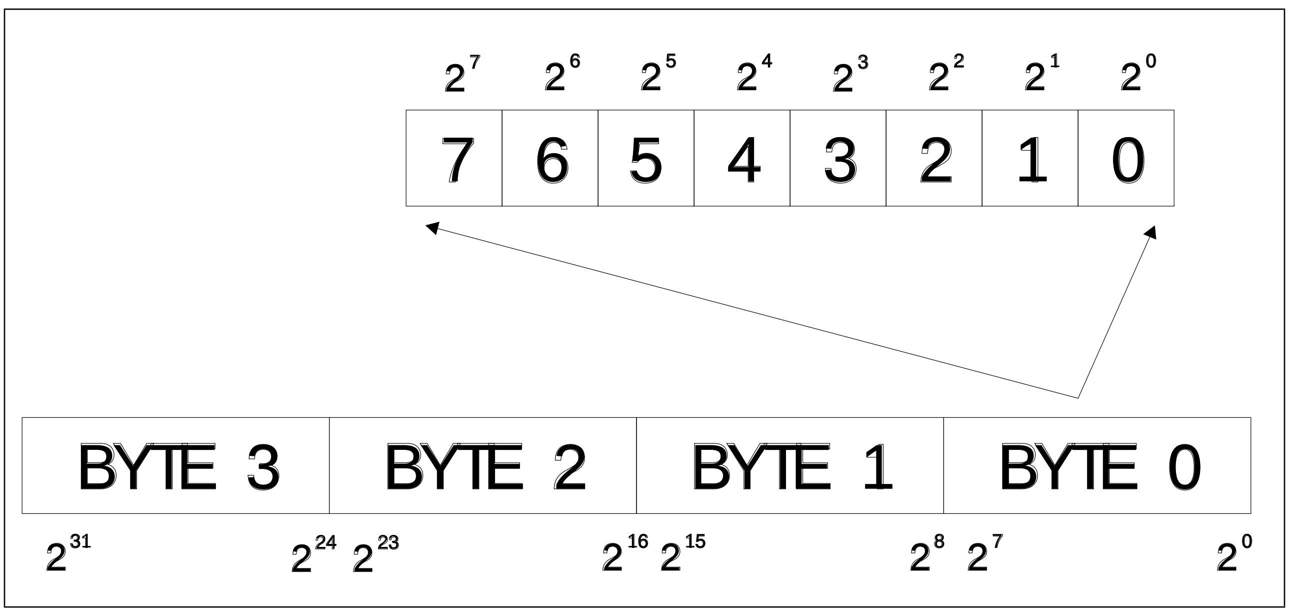


Figure 2. Bit and Byte Ordering

2.0 Serializer

The serializer is loaded with multiple pixels at the rising edge of the load clock. The pixels are then serialized internally to produce one pixel for each video clock cycle. The RGB640 features a programmable serializer capable of supporting a wide variety of frame buffer configurations.

As shown in Figure 3, four byte wide 16:1 multiplexers are used to form the raw pixel data. The raw pixel data is a 32 bit value that can be used to supply both the frame buffer and overlay pixel information. Two nibble wide 10:1 multiplexers are used to form the window ID busses.

Additionally, the serialization sequence can be varied from one horizontal line to the next. This feature can be used to provide enhanced video RAM update performance. This feature is referred to as pixel interleave support elsewhere in this document and is more fully explained in section section 2.4, “Pixel Interleave” on page 8.

2.1 Raw Pixel Data Select

The serializer produces four output bytes per pixel. Two bytes are dedicated to the frame buffer layer, the other two bytes can be used for either the frame buffer or overlay layer.

The four bytes of the raw pixel data are selected from pixel inputs 127:000 on the serializer interface. These 128 inputs are split into byte wide groups as given in Table 1 on page 6.

The serialization sequence can be independently defined for each of the four raw pixel bytes. Each raw pixel data byte has associated parameters that select the byte to use for the first serialized pixel and the number of bytes to skip over to find the byte for the next serialized pixel. These parameters are stored in the Serializer Raw Pixel Control Registers (registers 0x0002 to 0x0005).

For example, the start group parameter for raw pixel data (07:00) determines which of the 16 bytes on the

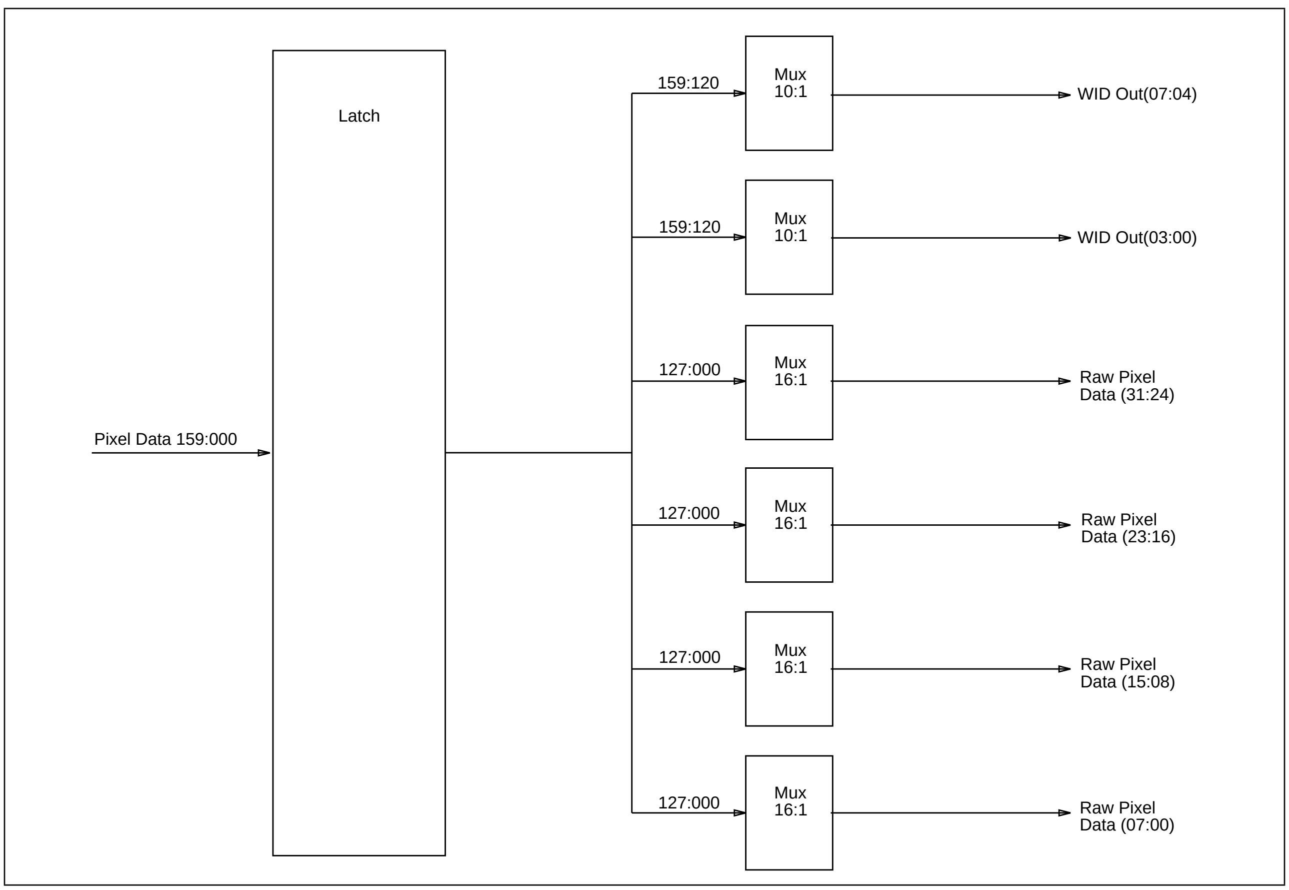


Figure 3. Serializer Data Flow

serializer interface is selected as the least significant byte of the first pixel (pixel A). The skip parameter determines how many bytes on the serializer to skip over to find the least significant byte of the next pixel (pixel B). The skip parameter is continually applied to find the least significant byte of pixels C, D, and so on. The sequence returns to the start group when all pixels have been serialized. This is determined by the Serializer Mode Register (register 0x0008). When the serializer is in 4:1 mode, for example, pixels A, B, C, and D are serialized before the sequence resets.

Table 1. Pixel Data Input Pin Groupings

Group	Inputs
0x0	007:000
0x1	015:008
0x2	023:016
0x3	031:024
0x4	039:032
0x5	047:040
0x6	055:048
0x7	063:056
0x8	071:064
0x9	079:072
0xA	087:080
0xB	095:088
0xC	103:096
0xD	111:104
0xE	119:112
0xF	127:120

Two of the four bytes of the raw pixel data are dedicated to the frame buffer layer, the other two bytes can be used for either the frame buffer layer or the overlay layer as shown in Figure 4.

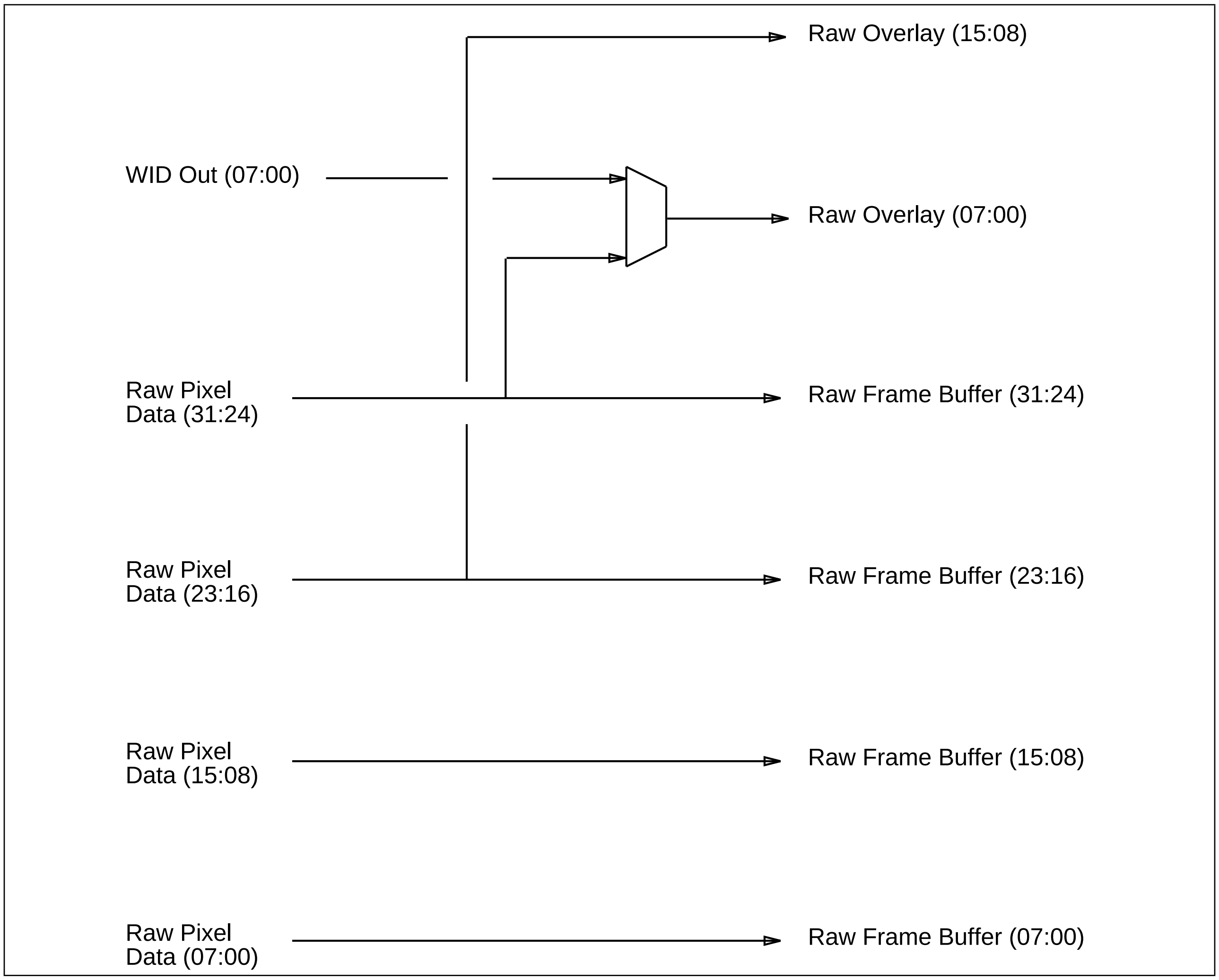


Figure 4. Raw Pixel Data Select

2.2 Window ID Select

The serializer also produces two additional nibbles per pixel. These two nibbles are used for window ID data, one for WID Out(07:04), the other for WID Out(03:00).

The window ID data is selected from pixel inputs 159:128 on the serializer interface. An extension into pixel inputs 127:120 is provided for use primarily in 5:1 serialization mode. These 40 inputs are split into nibble wide groups in Table 2.

Table 2. Window ID Input Pin Groupings

Group	Inputs
0x0	131:128
0x1	135:132
0x2	139:136
0x3	143:140
0x4	147:144
0x5	151:148
0x6	155:152
0x7	159:156
0x8	Not Available
0x9	Not Available
0xA	Not Available
0xB	Not Available
0xC	Not Available
0xD	Not Available
0xE	123:120
0xF	127:124

The serialization sequence can be independently defined for each of the two window ID nibbles, WID Out(07:04) and WID Out(03:00). Each of the two window ID nibbles has associated parameters that select the nibble to use for the first serialized window ID and the number of nibbles to skip over to find the nibble for the next serialized pixel. These parameters are stored in the Serializer WID Out Control Registers (registers 0x0006 and 0x0007).

For example, the start group parameter for window ID out (03:00) determines which of the 10 nibbles on the serializer interface is selected as the least significant nibble of the first window ID (window ID A). The skip parameter determines how many nibbles on the serializer to skip over to find the least significant nibble of the next window ID (window ID B). The skip parameter is continually applied to find the least significant nibble of window ID C, D, and so on. The sequence returns to the start group when all window ID's have been serialized. This is determined by the Serializer Mode Register (register 0x0008). When the serializer is in 4:1 mode, for example, window ID's A, B, C, and D are serialized before the sequence resets.

2.3 Common Serializer Configurations

Examples of common serializer configurations are given in tables 3, 4 and 5 on pages 9 through 11.

2.4 Pixel Interleave

Pixel interleave facilitates memory configurations that improve line drawing performance. This technique changes the serialization order on each horizontal line. The amount of rotation is programmable. The sequences are defined in tables 6, 7 and 8 on page 12. Pixel interleave is only available for 4:1, 5:1, and 8:1 multiplex modes. If the serializer is set to any other mode, then the Pixel Interleave Register (register 0x0009) must be set to 0x00 for proper operation.

2.5 Load Clock Interleave

Load clock interleave is used with 8 bit double buffered pixels ***in the the frame buffer layer only***. This technique swaps the positions of buffer A and buffer B after each load clock. Buffer A is in normal position for the first load clock on the first line through the fourth line. On the fifth line, buffer A is in swapped position for the first load clock. This pattern repeats every four lines. Table 9 on page 13 is a tabular description of load clock interleaving.

Load clock interleave can be activated only for 8 bit per pixel modes. In other modes, load clock interleave is disabled independent of the setting of the enable bit in the Miscellaneous Configuration Register (register 0x000A). Load clock interleave is referenced to the left side of the screen.

Table 3. Example Serializer Configurations

Serial Interface and Configuration Registers	Example 1 2:1 Mode	Example 2 2:1 Mode	Example 3 2:1 Mode	Example 4 2:1 Mode	Example 5 4:1 Mode	Example 6 4:1 Mode	Example 7 4:1 Mode
$PIX_{159:156}$					$WD_{07:04}$	$WD_{07:04}$	$WD_{03:00}$
$PIX_{155:152}$					$WD_{03:00}$	$WC_{07:04}$	$WC_{03:00}$
$PIX_{151:148}$			$WB_{07:04}$		$WC_{07:04}$	$WB_{07:04}$	$WB_{03:00}$
$PIX_{147:144}$			$WA_{07:04}$		$WC_{03:00}$	$WA_{07:04}$	$WA_{03:00}$
$PIX_{143:140}$	$WB_{07:04}$	$WB_{07:04}$		$WB_{03:00}$	$WB_{07:04}$	$WD_{03:00}$	$WD_{07:04}$
$PIX_{139:136}$	$WB_{03:00}$	$WA_{07:04}$		$WB_{07:04}$	$WB_{03:00}$	$WC_{03:00}$	$WC_{07:04}$
$PIX_{135:132}$	$WA_{07:04}$	$WB_{03:00}$	$WB_{03:00}$	$WA_{03:00}$	$WA_{07:04}$	$WB_{03:00}$	$WB_{07:04}$
$PIX_{131:128}$	$WA_{03:00}$	$WA_{03:00}$	$WA_{03:00}$	$WA_{07:04}$	$WA_{03:00}$	$WA_{03:00}$	$WA_{07:04}$
$PIX_{127:120}$					$D_{31:24}$	$D_{31:24}$	$D_{07:00}$
$PIX_{119:112}$					$D_{23:16}$	$C_{31:24}$	$C_{07:00}$
$PIX_{111:104}$			$B_{31:24}$		$D_{15:08}$	$B_{31:24}$	$B_{07:00}$
$PIX_{103:096}$			$A_{31:24}$		$D_{07:00}$	$A_{31:24}$	$A_{07:00}$
$PIX_{095:088}$					$C_{31:24}$	$D_{23:16}$	$D_{15:08}$
$PIX_{087:080}$					$C_{23:16}$	$C_{23:16}$	$C_{15:08}$
$PIX_{079:072}$			$B_{23:16}$		$C_{15:08}$	$B_{23:16}$	$B_{15:08}$
$PIX_{071:064}$			$A_{23:16}$		$C_{07:00}$	$A_{23:16}$	$A_{15:08}$
$PIX_{063:056}$	$B_{31:24}$	$B_{31:24}$		$B_{07:00}$	$B_{31:24}$	$D_{15:08}$	$D_{23:16}$
$PIX_{055:048}$	$B_{23:16}$	$A_{31:24}$		$B_{15:08}$	$B_{23:16}$	$C_{15:08}$	$C_{23:16}$
$PIX_{047:040}$	$B_{15:08}$	$B_{23:16}$	$B_{15:08}$	$B_{23:16}$	$B_{15:08}$	$B_{15:08}$	$B_{23:16}$
$PIX_{039:032}$	$B_{07:00}$	$A_{23:16}$	$A_{15:08}$	$B_{31:24}$	$B_{07:00}$	$A_{15:08}$	$A_{23:16}$
$PIX_{031:024}$	$A_{31:24}$	$B_{15:08}$		$A_{07:00}$	$A_{31:24}$	$D_{07:00}$	$D_{31:24}$
$PIX_{023:016}$	$A_{23:16}$	$A_{15:08}$		$A_{15:08}$	$A_{23:16}$	$C_{07:00}$	$C_{31:24}$
$PIX_{015:008}$	$A_{15:08}$	$B_{07:00}$	$B_{07:00}$	$A_{23:16}$	$A_{15:08}$	$B_{07:00}$	$B_{31:24}$
$PIX_{007:000}$	$A_{07:00}$	$A_{07:00}$	$A_{07:00}$	$A_{31:24}$	$A_{07:00}$	$A_{07:00}$	$A_{31:24}$
Register 0x0002	Start = 0x0 Skip = 0x3	Start = 0x0 Skip = 0x0	Start = 0x0 Skip = 0x0	Start = 0x3 Skip = 0x3	Start = 0x0 Skip = 0x3	Start = 0x0 Skip = 0x0	Start = 0xC Skip = 0x0
Register 0x0003	Start = 0x1 Skip = 0x3	Start = 0x2 Skip = 0x0	Start = 0x4 Skip = 0x0	Start = 0x2 Skip = 0x3	Start = 0x1 Skip = 0x3	Start = 0x4 Skip = 0x0	Start = 0x8 Skip = 0x0
Register 0x0004	Start = 0x2 Skip = 0x3	Start = 0x4 Skip = 0x0	Start = 0x8 Skip = 0x0	Start = 0x1 Skip = 0x3	Start = 0x2 Skip = 0x3	Start = 0x8 Skip = 0x0	Start = 0x4 Skip = 0x0
Register 0x0005	Start = 0x3 Skip = 0x3	Start = 0x6 Skip = 0x0	Start = 0xC Skip = 0x0	Start = 0x0 Skip = 0x3	Start = 0x3 Skip = 0x3	Start = 0xC Skip = 0x0	Start = 0x0 Skip = 0x0
Register 0x0006	Start = 0x0 Skip = 0x1	Start = 0x0 Skip = 0x0	Start = 0x0 Skip = 0x0	Start = 0x1 Skip = 0x1	Start = 0x0 Skip = 0x1	Start = 0x0 Skip = 0x0	Start = 0x4 Skip = 0x0
Register 0x0007	Start = 0x1 Skip = 0x1	Start = 0x2 Skip = 0x0	Start = 0x4 Skip = 0x0	Start = 0x0 Skip = 0x1	Start = 0x1 Skip = 0x1	Start = 0x4 Skip = 0x0	Start = 0x0 Skip = 0x0
Register 0x0008	Mode = 0x0	Mode = 0x0	Mode = 0x0	Mode = 0x0	Mode = 0x1	Mode = 0x1	Mode = 0x1

Notes:
WA = WID Out A, WB = WID Out B, etc.
A = Raw Pixel Data A, B = Raw Pixel Data B, etc.
N/A = Register setting has no effect.

Table 4. Example Serializer Configurations

Serial Interface and Configuration Registers	Example 8 8:1 Mode	Example 9 8:1 Mode	Example 10 8:1 Mode	Example 11 8:1 Mode	Example 12 16:1 Mode	Example 13 16:1 Mode	Example 14 5:1 Mode
$PIX_{159:156}$	$WH_{03:00}$	$WH_{03:00}$	$WH_{03:00}$	$WH_{03:00}$			$WE_{07:04}$
$PIX_{155:152}$	$WG_{03:00}$	$WG_{03:00}$	$WG_{03:00}$	$WG_{03:00}$			$WE_{03:00}$
$PIX_{151:148}$	$WF_{03:00}$	$WF_{03:00}$	$WF_{03:00}$	$WF_{03:00}$			$WD_{07:04}$
$PIX_{147:144}$	$WE_{03:00}$	$WE_{03:00}$	$WE_{03:00}$	$WE_{03:00}$			$WD_{03:00}$
$PIX_{143:140}$	$WD_{03:00}$	$WD_{03:00}$	$WD_{03:00}$	$WD_{03:00}$			$WC_{07:04}$
$PIX_{139:136}$	$WC_{03:00}$	$WC_{03:00}$	$WC_{03:00}$	$WC_{03:00}$			$WC_{03:00}$
$PIX_{135:132}$	$WB_{03:00}$	$WB_{03:00}$	$WB_{03:00}$	$WB_{03:00}$			$WB_{07:04}$
$PIX_{131:128}$	$WA_{03:00}$	$WA_{03:00}$	$WA_{03:00}$	$WA_{03:00}$			$WB_{03:00}$
$PIX_{127:120}$	$H_{15:08}$	$H_{15:08}$	$H_{31:24}$	$H_{07:00}$	$P_{07:00}$	$P_{31:24}$	$WA_{07:00}$
$PIX_{119:112}$	$H_{07:00}$	$G_{15:08}$	$G_{31:24}$	$H_{15:08}$	$O_{07:00}$	$O_{31:24}$	$E_{23:16}$
$PIX_{111:104}$	$G_{15:08}$	$F_{15:08}$	$F_{31:24}$	$G_{07:00}$	$N_{07:00}$	$N_{31:24}$	$D_{23:16}$
$PIX_{103:096}$	$G_{07:00}$	$E_{15:08}$	$E_{31:24}$	$G_{15:08}$	$M_{07:00}$	$M_{31:24}$	$C_{23:16}$
$PIX_{095:088}$	$F_{15:08}$	$D_{15:08}$	$D_{31:24}$	$F_{07:00}$	$L_{07:00}$	$L_{31:24}$	$B_{23:16}$
$PIX_{087:080}$	$F_{07:00}$	$C_{15:08}$	$C_{31:24}$	$F_{15:08}$	$K_{07:00}$	$K_{31:24}$	$A_{23:16}$
$PIX_{079:072}$	$E_{15:08}$	$B_{15:08}$	$B_{31:24}$	$E_{07:00}$	$J_{07:00}$	$J_{31:24}$	$E_{15:08}$
$PIX_{071:064}$	$E_{07:00}$	$A_{15:08}$	$A_{31:24}$	$E_{15:08}$	$I_{07:00}$	$I_{31:24}$	$D_{15:08}$
$PIX_{063:056}$	$D_{15:08}$	$H_{07:00}$	$H_{07:00}$	$D_{07:00}$	$H_{07:00}$	$H_{31:24}$	$C_{15:08}$
$PIX_{055:048}$	$D_{07:00}$	$G_{07:00}$	$G_{07:00}$	$D_{15:08}$	$G_{07:00}$	$G_{31:24}$	$B_{15:08}$
$PIX_{047:040}$	$C_{15:08}$	$F_{07:00}$	$F_{07:00}$	$C_{07:00}$	$F_{07:00}$	$F_{31:24}$	$A_{15:08}$
$PIX_{039:032}$	$C_{07:00}$	$E_{07:00}$	$E_{07:00}$	$C_{15:08}$	$E_{07:00}$	$E_{31:24}$	$E_{07:00}$
$PIX_{031:024}$	$B_{15:08}$	$D_{07:00}$	$D_{07:00}$	$B_{07:00}$	$D_{07:00}$	$D_{31:24}$	$D_{07:0}$
$PIX_{023:016}$	$B_{07:00}$	$C_{07:00}$	$C_{07:00}$	$B_{15:08}$	$C_{07:00}$	$C_{31:24}$	$C_{07:0}$
$PIX_{015:008}$	$A_{15:08}$	$B_{07:00}$	$B_{07:00}$	$A_{07:00}$	$B_{07:00}$	$B_{31:24}$	$B_{07:0}$
$PIX_{007:000}$	$A_{07:00}$	$A_{07:00}$	$A_{07:00}$	$A_{15:08}$	$A_{07:00}$	$A_{31:24}$	$A_{07:0}$
Register 0x0002	Start = 0x0 Skip = 0x1	Start = 0x0 Skip = 0x0	Start = 0x0 Skip = 0x0	Start = 0x1 Skip = 0x1	Start = 0x0 Skip = 0x0	Start = N/A Skip = N/A	Start = 0x0 Skip = 0x0
Register 0x0003	Start = 0x1 Skip = 0x1	Start = 0x8 Skip = 0x0	Start = N/A Skip = N/A	Start = 0x0 Skip = 0x1	Start = N/A Skip = N/A	Start = N/A Skip = N/A	Start = 0x5 Skip = 0x0
Register 0x0004	Start = N/A Skip = N/A	Start = N/A Skip = N/A	Start = N/A Skip = N/A	Start = N/A Skip = N/A	Start = N/A Skip = N/A	Start = N/A Skip = N/A	Start = 0xA Skip = 0x0
Register 0x0005	Start = N/A Skip = N/A	Start = N/A Skip = N/A	Start = 0x8 Skip = 0x0	Start = N/A Skip = N/A	Start = N/A Skip = N/A	Start = 0x0 Skip = 0x0	Start = N/A Skip = N/A
Register 0x0006	Start = 0x0 Skip = 0x0	Start = 0x0 Skip = 0x0	Start = 0x0 Skip = 0x0	Start = 0x0 Skip = 0x0	Start = N/A Skip = N/A	Start = N/A Skip = N/A	Start = 0xE Skip = 0x1
Register 0x0007	Start = N/A Skip = N/A	Start = N/A Skip = N/A	Start = N/A Skip = N/A	Start = N/A Skip = N/A	Start = N/A Skip = N/A	Start = N/A Skip = N/A	Start = 0xF Skip = 0x1
Register 0x0008	Mode = 0x2	Mode = 0x2	Mode = 0x2	Mode = 0x2	Mode = 0x3	Mode = 0x3	Mode = 0x6

Notes:
WA = WID Out A, WB = WID Out B, etc.
A = Raw Pixel Data A, B = Raw Pixel Data B, etc.
N/A = Register setting has no effect.

Table 5. Example Serializer Configurations

Serial Interface and Configuration Registers	Example 15 5:1 Mode	Example 16 16:3 Mode			Example 17 16:3 Mode		
		1st Load Clock	2nd Load Clock	3rd Load Clock	1st Load Clock	2nd Load Clock	3rd Load Clock
$PIX_{159:156}$	$WE_{07:04}$						
$PIX_{155:152}$	$WE_{03:00}$						
$PIX_{151:148}$	$WD_{07:04}$						
$PIX_{147:144}$	$WD_{03:00}$						
$PIX_{143:140}$	$WC_{07:04}$						
$PIX_{139:136}$	$WC_{03:00}$						
$PIX_{135:132}$	$WB_{07:04}$						
$PIX_{131:128}$	$WB_{03:00}$						
$PIX_{127:120}$	$WA_{07:00}$	$F_{07:00}$	$K_{15:08}$	$P_{23:16}$	$F_{23:16}$	$K_{15:08}$	$P_{07:00}$
$PIX_{119:112}$	$E_{23:16}$	$E_{23:16}$	$K_{07:00}$	$P_{15:08}$	$E_{07:00}$	$K_{23:16}$	$P_{15:08}$
$PIX_{111:104}$	$E_{15:08}$	$E_{15:08}$	$J_{23:16}$	$P_{07:00}$	$E_{15:08}$	$J_{07:00}$	$P_{23:16}$
$PIX_{103:096}$	$E_{07:00}$	$E_{07:00}$	$J_{15:08}$	$O_{23:16}$	$E_{23:16}$	$J_{15:08}$	$O_{07:00}$
$PIX_{095:088}$	$D_{23:16}$	$D_{23:16}$	$J_{07:00}$	$O_{15:08}$	$D_{07:00}$	$J_{23:16}$	$O_{15:08}$
$PIX_{087:080}$	$D_{15:08}$	$D_{15:08}$	$I_{23:16}$	$O_{07:00}$	$D_{15:08}$	$I_{07:00}$	$O_{23:16}$
$PIX_{079:072}$	$D_{07:00}$	$D_{07:00}$	$I_{15:08}$	$N_{23:16}$	$D_{23:16}$	$I_{15:08}$	$N_{07:00}$
$PIX_{071:064}$	$C_{23:16}$	$C_{23:16}$	$I_{07:00}$	$N_{15:08}$	$C_{07:00}$	$I_{23:16}$	$N_{15:08}$
$PIX_{063:056}$	$C_{15:08}$	$C_{15:08}$	$H_{23:16}$	$N_{07:00}$	$C_{15:08}$	$H_{07:00}$	$N_{23:16}$
$PIX_{055:048}$	$C_{07:00}$	$C_{07:00}$	$H_{15:08}$	$M_{23:16}$	$C_{23:16}$	$H_{15:08}$	$M_{07:00}$
$PIX_{047:040}$	$B_{23:16}$	$B_{23:16}$	$H_{07:00}$	$M_{15:08}$	$B_{07:00}$	$H_{23:16}$	$M_{15:08}$
$PIX_{039:032}$	$B_{15:08}$	$B_{15:08}$	$G_{23:16}$	$M_{07:00}$	$B_{15:08}$	$G_{07:00}$	$M_{23:16}$
$PIX_{031:024}$	$B_{07:00}$	$B_{07:00}$	$G_{15:08}$	$L_{23:16}$	$B_{23:16}$	$G_{15:08}$	$L_{07:00}$
$PIX_{023:016}$	$A_{23:16}$	$A_{23:16}$	$G_{07:00}$	$L_{15:08}$	$A_{07:00}$	$G_{23:16}$	$L_{15:08}$
$PIX_{015:008}$	$A_{15:08}$	$A_{15:08}$	$F_{23:16}$	$L_{07:00}$	$A_{15:08}$	$F_{07:00}$	$L_{23:16}$
$PIX_{007:000}$	$A_{07:00}$	$A_{07:00}$	$F_{15:08}$	$K_{23:16}$	$A_{23:16}$	$F_{15:08}$	$K_{07:00}$
Register 0x0002	Start = 0x0 Skip = 0x2	Start = 0x0 Skip = 0x2			Start = 0x2 Skip = 0x2		
Register 0x0003	Start = 0x1 Skip = 0x2	Start = 0x1 Skip = 0x2			Start = 0x1 Skip = 0x2		
Register 0x0004	Start = 0x2 Skip = 0x2	Start = 0x2 Skip = 0x2			Start = 0x0 Skip = 0x2		
Register 0x0005	Start = N/A Skip = N/A	Start = N/A Skip = N/A			Start = N/A Skip = N/A		
Register 0x0006	Start = 0xE Skip = 0x1	Start = N/A Skip = N/A			Start = N/A Skip = N/A		
Register 0x0007	Start = 0xF Skip = 0x1	Start = N/A Skip = N/A			Start = N/A Skip = N/A		
Register 0x0008	Mode = 0x6	Mode = 0x5			Mode = 0x5		

Notes:
WA = WID Out A, WB = WID Out B, etc.
A = Raw Pixel Data A, B = Raw Pixel Data B, etc.
N/A = Register setting has no effect.

Table 6. 4:1 Pixel Interleave

Line	Interleave 0	Interleave 1	Interleave 2	Interleave 3
1	A B C D	A B C D	A B C D	A B C D
2	A B C D	B C D A	C D A B	D A B C
3	A B C D	C D A B	A B C D	C D A B
4	A B C D	D A B C	C D A B	B C D A

Table 7. 5:1 Pixel Interleave

Line	Interleave 0	Interleave 1	Interleave 2	Interleave 3	Interleave 4
1	A B C D E	A B C D E	A B C D E	A B C D E	A B C D E
2	A B C D E	B C D E A	C D E A B	D E A B C	E A B C D
3	A B C D E	C D E A B	E A B C D	B C D E A	D E A B C
4	A B C D E	D E A B C	B C D E A	E A B C D	C D EA B
5	A B C D E	E A B C D	D E A B C	C D E A B	B C D E A

Table 8. 8:1 Pixel Interleave

Line	Interleave 0	Interleave 1	Interleave 2	Interleave 3
1	A B C D E F G H	A B C D E F G H	A B C D E F G H	A B C D E F G H
2	A B C D E F G H	B C D E F G H A	C D E F G H A B	D E F G H A B C
3	A B C D E F G H	C D E F G H A B	E F G H A B C D	G H A B C D E F
4	A B C D E F G H	D E F G H A B C	G H A B C D E F	B C D E F G H A
5	A B C D E F G H	E F G H A B C D	A B C D E F G H	E F G H A B C D
6	A B C D E F G H	F G H A B C D E	C D E F G H A B	H A B C D E F G
7	A B C D E F G H	G H A B C D E F	E F G H A B C D	C D E F G H A B
8	A B C D E F G H	H A B C D E F G	G H A B C D E F	F G H A B C D E

Line	Interleave 4	Interleave 5	Interleave 6	Interleave 7
1	A B C D E F G H	A B C D E F G H	A B C D E F G H	A B C D E F G H
2	E F G H A B C D	F G H A B C D E	G H A B C D E F	H A B C D E F G
3	A B C D E F G H	C D E F G H A B	E F G H A B C D	G H A B C D E F
4	E F G H A B C D	H A B C D E F G	C D E F G H A B	F G H A B C D E
5	A B C D E F G H	E F G H A B C D	A B C D E F G H	E F G H A B C D
6	E F G H A B C D	B C D E F G H A	G H A B C D E F	D E F G H A B C
7	A B C D E F G H	G H A B C D E F	E F G H A B C D	C D E F G H A B
8	E F G H A B C D	D E F G H A B C	C D E F G H A B	B C D E F G H A

Table 9. Load Clock Interleave

Line	Load Clock 0	Load Clock 1	Load Clock 2	Load Clock 3
1	Normal	Swapped	Normal	Etc.
2	Normal	Swapped	Normal	Etc.
3	Normal	Swapped	Normal	Etc.
4	Normal	Swapped	Normal	Etc.
5	Swapped	Normal	Swapped	Etc.
6	Swapped	Normal	Swapped	Etc.
7	Swapped	Normal	Swapped	Etc.
8	Swapped	Normal	Swapped	Etc.
9	Normal	Swapped	Normal	Etc.
10	Etc.	Etc.	Etc.	Etc.

3.0 Pixel Formats

This section presents a complete list of the pixel types supported by the RGB640. Not all pixel types are supported for both the overlay and the frame buffer layer. The pixel type for the overlay layer and frame buffer layer is determined by their respective window attribute table.

Depths of 4, 8, 10, 16, 24, and 30 bits per pixel are supported by the RGB640. The pixel types can be palette-indexed, palette-bypassed, gray scale, YUV, or IRGB. Palette-bypassed types are linearized to ten bits by copying some of the most significant bits of the pixel to the missing least significant bits of the result.

3.1 Palette-Indexed Formats

Seven palette-indexed pixel types are available. Palette index calculation is detailed in Table 10 on page 18. Multiple palettes within the 1K look up table can be partitioned using different start addresses for differing windows. The palette start address is designated by S(03) to S(00), with S(03) representing the most significant bit and S(00) representing the least significant bit. The palette start address can vary from window to window and is determined by byte 1 of the frame buffer or overlay window attribute table. The pixel bits are designated P(29) to P(00), with P(29) representing the most significant bit and P(00) representing the least significant bit. These pixel values are the result of serialization and are supplied from the selected frame buffer data bus or the selected overlay data bus (as shown in Figure 1 on page 3). The calculation of the palette address is shown in Table 10 on page 18.

3.2 Palette-Bypassed Formats

Several palette-bypassed pixel types are available. Each linearizes any missing bits in the 10 bit expanded result by copying some of the most significant bits of the pixel to the missing bits of the bypassed result. The pixel bits are designated P(29) to P(00), with P(29) representing the most significant bit and P(00) representing the least significant bit. These pixel values are the result of serialization and are supplied from the selected frame buffer data bus or the selected overlay data bus. The resulting bits of the overlay bypass data bus or the frame buffer bypass data bus (Figure 1 on page 3) are shown in Table 11 on page 20.

3.3 YUV Formats

Three YUV formats are supported - 15 bit subsampled, 16 bit subsampled, and 24 bit. The subsampled formats provide intensity information (Y) and half of the color information (U or V) with each pixel. The 24 bit format provides all intensity and color information for each pixel.

The subsampled modes have two alignment formats that can be selected with the frame buffer window attribute table. Odd pixels can be defined as either a YU or YV combination. Regardless of the format, the YU pixel is to the left of the YV pixel in a YU/YV pixel pair. Synchronization starts at the left edge of the screen. The first pixel is defined as pixel zero, an even pixel.

For the 15 bit subsampled YUV format, the next most significant bit controls how the pixel is to be displayed. This bit allows the pixel to be either the 15 bit subsampled YUV format or the 5/5/5 RGB palette-indexed format. The definition of this select bit is programmable with the frame buffer window attribute table by selecting the proper PIXTYP in byte 0.

Tables 12 through 16 on page 21 show how the pixel bits are formatted to produce YUV data (refer to Figure 1 on page 3). The pixel bits are designated P(29) to P(00), with P(29) representing the most significant bit and P(00) representing the least significant bit. These pixel values are the result of serialization and are supplied from the selected frame buffer data bus.

3.3.1 Color Space Conversion

The formatted YUV data are converted to RGB values using the following equations:

$$Red = Y + K1 \times (V - 128)$$
$$Green = Y - K2 \times (V - 128) - K3 \times (U - 128)$$
$$Blue = Y + K4 \times (U - 128)$$

The constants K1, K2, K3 and K4 are supplied using the registers at indices 0x0070 through 0x0073.

3.3.2 Input Values

Y, the luminance value, ranges from 0 (0x00) to 255 (0xff). U and V, the chrominance values, are “offset 128” values, and represent the range -128 to +127 after 128 (0x80) is subtracted. (This makes an incoming value of 128 the “zero chrominance” value.)

For 24-bit and 16-bit YUV there are 256 values for Y, U and V. For the 15-bit YUV format U and V that have a 7-

bit field will only have 128 values. As shown in tables 15 and 16, the 7-bit field is missing the low order bit, and a '0' is substituted internally for this bit. Note that the most positive value for U and V in this case is 0xfe.

3.3.3 Range Endpoints

Some encoding systems for video do not use the full scale range. E.g., it is common to restrict the range of Y values to 16 through 235. The unused values may be used for other purposes. In particular, the values 0 and 255 are used for timing information.

Some color space converters convert values less than 16 to 16, and values greater than 235 to 235. The RGB640 does *not* do this. All values of Y, U and V are presented unchanged to the color space converter as received from the frame buffer.

3.3.4 Coefficients

The constants K1 - K4 are 8 bit values are can be programmed by the user with four 8 bit registers. The constants represent the range 0.0 to 1.9921875, using the following bit weighting:

Kn[7] = 1.0000000 (MSB)

Kn[6] = 0.5000000

Kn[5] = 0.2500000

Kn[4] = 0.1250000

Kn[3] = 0.0625000

Kn[2] = 0.0312500

Kn[1] = 0.0156250

Kn[0] = 0.0078125 (LSB)

(Kn is the register K1, K2, K3 or K4. [7], [6], [5]… is the bit position within the register.)

The constants must be stable during conversion, (i.e., the register values must not be changed.)

3.3.5 Bit Precision and Output Values

Internally, the multiplications of the U and V values times the K constants are generated at full 16-bit precision. When these intermediate results are added to the Y values "saturated" arithmetic is used: attempted overflows beyond 16 bits saturated at all '1's, and attempted underflows saturate at all '0's.

The resulting 16-bit values are truncated to 10 bits. These results range in value from 0.0 to 255.75. Since the RGB values are usually regarded as integer values ranging from 0 through 1023, one can regard the YUV to RGB conversion as having the R,G,B values each implicitly multiplied by 4. No shifting takes place – the binary point is simply shifted right by 2 places.

3.3.6 Interpolation

For the sub-sampled formats, the missing chrominance values will be generated by simply duplicating the existing value from the sample pair. Thus, for the incoming samples:

[Y0,U0] [Y1,V0] [Y2,U2] [Y3,V2] …

the pixels presented to the color space converter will be:

[Y0,U0,V0] [Y1,U0,V0] [Y2,U2,V2] [Y3,U2,V2] …

3.3.7 Edge Effect Handling

With mixed YUV and RGB pixels the YUV pixels must occur in pairs. If this condition is violated (there is a pair of pixels with one pixel tagged as RGB and the other tagged as YUV) then the YUV pixel will be missing either the U or the V value. When this happens the value of the missing U or V value that is input to the color space converter is indeterminate, and therefor the generated RGB value will be indeterminate.

As an example of this situation, consider a GUI environment in which a rectangular window of subsampled YUV pixels is displayed against a background of RGB pixels. If the width of the YUV window is an odd number of pixels, then the rightmost pixels of that window will be missing their U or V values. Whatever values are used for the missing U and V values will distort the colors of the right edge, hence the phrase "edge effect".

3.4 IRGB Format

The overlay bypass channel provides IRGB expansion that was previously available in several VGA modes. Table 17 on page 22 shows how the pixel bits are translated to form 30 bits of data on the overlay bypass data bus (refer to Figure 1 on page 3). The pixel bits are designated P(03) to P(00), with P(03) representing the most significant bit and P(00) representing the least significant bit. These pixel values are the result of serialization and are supplied from the selected overlay data bus.

Table 10. Palette-Indexed Pixel Formats Selectable Via Window Attribute Tables

4 Bit Common Palette Index											
Red		S(03)	S(02)	S(01)	S(00)	0	0	0	0	0	0
	+	0	0	0	0	0	0	P(03)	P(02)	P(01)	P(00)
Green		S(03)	S(02)	S(01)	S(00)	0	0	0	0	0	0
	+	0	0	0	0	0	0	P(03)	P(02)	P(01)	P(00)
Blue		S(03)	S(02)	S(01)	S(00)	0	0	0	0	0	0
	+	0	0	0	0	0	0	P(03)	P(02)	P(01)	P(00)
8 Bit Common Palette Index											
Red		S(03)	S(02)	S(01)	S(00)	0	0	0	0	0	0
	+	0	0	P(07)	P(06)	P(05)	P(04)	P(03)	P(02)	P(01)	P(00)
Green		S(03)	S(02)	S(01)	S(00)	0	0	0	0	0	0
	+	0	0	P(07)	P(06)	P(05)	P(04)	P(03)	P(02)	P(01)	P(00)
Blue		S(03)	S(02)	S(01)	S(00)	0	0	0	0	0	0
	+	0	0	P(07)	P(06)	P(05)	P(04)	P(03)	P(02)	P(01)	P(00)
3/3/2 RGB Palette Index											
Red		S(03)	S(02)	S(01)	S(00)	0	0	0	0	0	0
	+	0	0	0	0	0	0	0	P(07)	P(06)	P(05)
Green		S(03)	S(02)	S(01)	S(00)	0	0	0	0	0	0
	+	0	0	0	0	0	0	0	P(04)	P(03)	P(02)
Blue		S(03)	S(02)	S(01)	S(00)	0	0	0	0	0	0
	+	0	0	0	0	0	0	0	0	P(01)	P(00)
5/6/5 RGB Palette Index											
Red		S(03)	S(02)	S(01)	S(00)	0	0	0	0	0	0
	+	0	0	0	0	0	P(15)	P(14)	P(13)	P(12)	P(11)
Green		S(03)	S(02)	S(01)	S(00)	0	0	0	0	0	0
	+	0	0	0	0	P(10)	P(09)	P(08)	P(07)	P(06)	P(05)
Blue		S(03)	S(02)	S(01)	S(00)	0	0	0	0	0	0
	+	0	0	0	0	0	P(04)	P(03)	P(02)	P(01)	P(00)

Table 10. Palette-Indexed Pixel Formats Selectable Via Window Attribute Tables (Continued)

5/5/5 RGB Palette Index											
Red		S(03)	S(02)	S(01)	S(00)	0	0	0	0	0	0
	+	0	0	0	0	0	P(14)	P(13)	P(12)	P(11)	P(10)
Green		S(03)	S(02)	S(01)	S(00)	0	0	0	0	0	0
	+	0	0	0	0	0	P(09)	P(08)	P(07)	P(06)	P(05)
Blue		S(03)	S(02)	S(01)	S(00)	0	0	0	0	0	0
	+	0	0	0	0	0	P(04)	P(03)	P(02)	P(01)	P(00)
8/8/8 RGB Palette Index											
Red		S(03)	S(02)	S(01)	S(00)	0	0	0	0	0	0
	+	0	0	P(23)	P(22)	P(21)	P(20)	P(19)	P(18)	P(17)	P(16)
Green		S(03)	S(02)	S(01)	S(00)	0	0	0	0	0	0
	+	0	0	P(15)	P(14)	P(13)	P(12)	P(11)	P(10)	P(09)	P(08)
Blue		S(03)	S(02)	S(01)	S(00)	0	0	0	0	0	0
	+	0	0	P(07)	P(06)	P(05)	P(04)	P(03)	P(02)	P(01)	P(00)
10/10/10 RGB Palette Index											
Red		P(29)	P(28)	P(27)	P(26)	P(25)	P(24)	P(23)	P(22)	P(21)	P(20)
Green		P(19)	P(18)	P(17)	P(16)	P(15)	P(14)	P(13)	P(12)	P(11)	P(10)
Blue		P(09)	P(08)	P(07)	P(06)	P(05)	P(04)	P(03)	P(02)	P(01)	P(00)

Table 11. Palette-Bypassed Pixel Formats Selectable Via Window Attribute Tables

	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
4 Bit Gray Scale										
Red	P(03)	P(02)	P(01)	P(00)	P(03)	P(02)	P(01)	P(00)	P(03)	P(02)
Green	P(03)	P(02)	P(01)	P(00)	P(03)	P(02)	P(01)	P(00)	P(03)	P(02)
Blue	P(03)	P(02)	P(01)	P(00)	P(03)	P(02)	P(01)	P(00)	P(03)	P(02)
3/3/2 RGB Palette Bypass										
Red	P(07)	P(06)	P(05)	P(07)	P(06)	P(05)	P(07)	P(06)	P(05)	P(07)
Green	P(04)	P(03)	P(02)	P(04)	P(03)	P(02)	P(04)	P(03)	P(02)	P(04)
Blue	P(01)	P(00)	P(01)	P(00)	P(01)	P(00)	P(01)	P(00)	P(01)	P(00)
8 Bit Gray Scale										
Red	P(07)	P(06)	P(05)	P(04)	P(03)	P(02)	P(01)	P(00)	P(07)	P(06)
Green	P(07)	P(06)	P(05)	P(04)	P(03)	P(02)	P(01)	P(00)	P(07)	P(06)
Blue	P(07)	P(06)	P(05)	P(04)	P(03)	P(02)	P(01)	P(00)	P(07)	P(06)
5/6/5 RGB Palette Bypass										
Red	P(15)	P(14)	P(13)	P(12)	P(11)	P(15)	P(14)	P(13)	P(12)	P(11)
Green	P(10)	P(09)	P(08)	P(07)	P(06)	P(05)	P(10)	P(09)	P(08)	P(07)
Blue	P(04)	P(03)	P(02)	P(01)	P(00)	P(04)	P(03)	P(02)	P(01)	P(00)
10 Bit Gray Scale										
Red	P(09)	P(08)	P(07)	P(06)	P(05)	P(04)	P(03)	P(02)	P(01)	P(00)
Green	P(09)	P(08)	P(07)	P(06)	P(05)	P(04)	P(03)	P(02)	P(01)	P(00)
Blue	P(09)	P(08)	P(07)	P(06)	P(05)	P(04)	P(03)	P(02)	P(01)	P(00)
8/8/8 RGB Palette Bypass										
Red	P(23)	P(22)	P(21)	P(20)	P(19)	P(18)	P(17)	P(16)	P(23)	P(22)
Green	P(15)	P(14)	P(13)	P(12)	P(11)	P(10)	P(09)	P(08)	P(15)	P(14)
Blue	P(07)	P(06)	P(05)	P(04)	P(03)	P(02)	P(01)	P(00)	P(07)	P(06)
10/10/10 RGB Palette Bypass										
Red	P(29)	P(28)	P(27)	P(26)	P(25)	P(24)	P(23)	P(22)	P(21)	P(20)
Green	P(19)	P(18)	P(17)	P(16)	P(15)	P(14)	P(13)	P(12)	P(11)	P(10)
Blue	P(09)	P(08)	P(07)	P(06)	P(05)	P(04)	P(03)	P(02)	P(01)	P(00)

Table 12. 24 Bit YUV Pixel Format

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Y	P(23)	P(22)	P(21)	P(20)	P(19)	P(18)	P(17)	P(16)
U	P(15)	P(14)	P(13)	P(12)	P(11)	P(10)	P(09)	P(08)
V	P(07)	P(06)	P(05)	P(04)	P(03)	P(02)	P(01)	P(00)

Table 13. 16 Bit YUV Pixel Format (ODD Bit Of Frame Buffer WAT Byte 0 = 0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Even Pixels								
Y	P(15)	P(14)	P(13)	P(12)	P(11)	P(10)	P(09)	P(08)
U	P(07)	P(06)	P(05)	P(04)	P(03)	P(02)	P(01)	P(00)
Odd Pixels								
Y	P(15)	P(14)	P(13)	P(12)	P(11)	P(10)	P(09)	P(08)
V	P(07)	P(06)	P(05)	P(04)	P(03)	P(02)	P(01)	P(00)

Table 14. 16 Bit YUV Pixel Format (ODD Bit Of Frame Buffer WAT Byte 0 = 1)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Odd Pixels								
Y	P(15)	P(14)	P(13)	P(12)	P(11)	P(10)	P(09)	P(08)
U	P(07)	P(06)	P(05)	P(04)	P(03)	P(02)	P(01)	P(00)
Even Pixels								
Y	P(15)	P(14)	P(13)	P(12)	P(11)	P(10)	P(09)	P(08)
V	P(07)	P(06)	P(05)	P(04)	P(03)	P(02)	P(01)	P(00)

Table 15. 15 Bit YUV Pixel Format (ODD Bit Of Frame Buffer WAT Byte 0 = 0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Even Pixels								
Y	P(14)	P(13)	P(12)	P(11)	P(10)	P(09)	P(08)	P(07)
U	P(06)	P(05)	P(04)	P(03)	P(02)	P(01)	P(00)	0
Odd Pixels								
Y	P(14)	P(13)	P(12)	P(11)	P(10)	P(09)	P(08)	P(07)
V	P(06)	P(05)	P(04)	P(03)	P(02)	P(01)	P(00)	0

Table 16. 15 Bit YUV Pixel Format (ODD Bit Of Frame Buffer WAT Byte 0 = 1)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Odd Pixels								
Y	P(14)	P(13)	P(12)	P(11)	P(10)	P(09)	P(08)	P(07)
U	P(06)	P(05)	P(04)	P(03)	P(02)	P(01)	P(00)	0
Even Pixels								
Y	P(14)	P(13)	P(12)	P(11)	P(10)	P(09)	P(08)	P(07)
V	P(06)	P(05)	P(04)	P(03)	P(02)	P(01)	P(00)	0

Table 17. IRGB Pixel Formatt

P(03:00)	Color	Red(09:00)	Green(09:00)	Blue(09:00)
0000	Black	0x000	0x000	0x000
0001	Blue	0x000	0x000	0x2AA
0010	Green	0x000	0x2AA	0x000
0011	Cyan	0x000	0x2AA	0x2AA
0100	Red	0x2AA	0x000	0x000
0101	Magenta	0x2AA	0x000	0x2AA
0110	Brown	0x2AA	0x155	0x000
0111	White	0x2AA	0x2AA	0x2AA
1000	Gray	0x155	0x155	0x155
1001	Light Blue	0x155	0x155	0x3FF
1010	Light Green	0x155	0x3FF	0x155
1011	Light Cyan	0x155	0x3FF	0x3FF
1100	Ligh Red	0x3FF	0x155	0x155
1101	Light Magenta	0x3FF	0x155	0x3FF
1110	Yellow	0x3FF	0x3FF	0x155
1111	Bright White	0x3FF	0x3FF	0x3FF

4.0 Window Attribute Tables

The RGB640 has two independent window attribute tables, one for the frame buffer layer and one for the overlay layer. The Frame Buffer Window Attribute Table controls the display mode of the frame buffer data, selects the frame buffer palette, and selects the buffer for double buffered pixels. The Overlay Window Attribute Table controls the display mode of the overlay data, selects the overlay palette, and selects the buffer for double buffered overlay pixels. In addition, the overlay window attribute table controls the blending of the frame buffer and overlay layers.

Frame buffer and overlay window attribute entries may be selected by a common window ID from the serializer or by two independent window ID's from the serializer.

Note that, in blend mode, both frame buffer and overlay layers can be programmed to use the palette. There is, however, only one palette resource. This results in an error condition. If both layers try to use the palette, the frame buffer will gain access to the palette and the overlay data will be forced through the bypass channel.

4.1 Frame Buffer Window Attribute Table

4.1.1 Byte 0: Pixel Type

7	6	5	4	3	2	1	0
RESERVED				PIXTYP			

POR: All Byte 0 entries power on to 0x00

Bits 7 - 4: RESERVED

Bits 3 - 0: PIXTYP -Pixel Type. Selects the format used to display frame buffer data.

PIXTYP	Bits Per Pixel	Pixel Format
0000	8	3/3/2 RGB Palette Index
0001	8	3/3/2 RGB Palette Bypass
0010	8	8 Bit Gray Scale
0011	8	8 Bit Common Palette Index
0100	16	5/6/5 RGB Palette Index
0101	16	5/6/5 RGB Palette Bypass
0110	10	10 Bit Gray Scale
0111	16	16 Bit YUV
1000	24	8/8/8 RGB Palette Index
1001	24	8/8/8 RGB Palette Bypass
1010		Reserved
1011	24	24 Bit YUV
1100	30	10/10/10 RGB Palette Index
1101	30	10/10/10 RGB Palette Bypass
1110	15	5/5/5 RGB Palette Index / 15 Bit YUV (MSB = 0 for RGB)
1111	15	5/5/5 RGB Palette Index / 15 Bit YUV (MSB = 1for RGB)

4.1.2 Byte 1: Palette Start Address

7	6	5	4	3	2	1	0
RESERVED				START			

POR: All Byte 1 entries power on to 0x00

Bits 7 - 4: RESERVED

Bits 3 - 0: START - Palette Start Address. This field is left shifted six bits and added to the pixel data to form a ten bit palette address for the palette-indexed formats. Refer to Table 10 on page 18 for details on the calculation of the palette address.

4.1.3 Byte 2: Buffer Select

7	6	5	4	3	2	1	0
RESERVED							SEL

POR: All Byte 2 entries power on to 0x00

Bits 7 - 1: RESERVED

Bit 0: SEL - Buffer Select. Chooses buffer A or buffer B for double buffered frame buffer data formats.

0 Buffer A Selected

1 Buffer B Selected

The bits selected are given below:

Bits Per Pixel	Buffer A Selected (SEL = 0)	Buffer B Selected (SEL = 1)
8	Raw Pixel Data(07:00)	Raw Pixel Data(15:08)
16	Raw Pixel Data(15:00)	Raw Pixel Data(31:16)

4.1.4 Byte 3: Control Bits

7	6	5	4	3	2	1	0
RESERVED						ODD	XDE

POR: All Byte 3 entries power on to 0x00

Bits 7 - 2: RESERVED

Bit 1: ODD - Odd/Even Select. Controls pixel alignment for 15 and 16 bit YUV formats.

0 Even pixels contain YU information and odd pixels contain YV information.

1 Odd pixels contain YU information and even pixels contain YV information.

Synchronization starts at the left edge of the screen. The first pixel is defined as pixel zero.

Bit 0: XDE - Crosshair Display Enable. Controls whether the crosshair is visible.

0 Crosshair display is disabled

1 Crosshair display is enabled

4.2 Overlay Window Attribute Table

4.2.1 Byte 0: Pixel Type

7	6	5	4	3	2	1	0
RESERVED					PIXTYP		

POR: All Byte 0 entries power on to 0x00

Bits 7 - 3: RESERVED

Bits 2- 0: PIXTYP - Pixel Type. Selects the format used to display overlay data. ***The alternate format listed in the following table is used during the error condition when layer blend is selected and both the overlay layer and the frame buffer layer are requesting the palette.***

PIXTYP	Bits Per Pixel	Pixel Format	Alternate Format
000	4	4 Bit Common Palette Index	IRGB Expansion
001	4	IRGB Expansion	
010		Reserved	
011	4	4 Bit Gray Scale	
100	8	8 Bit Common Palette Index	3/3/2 RGB Palette Bypass
101	8	3/3/2 RGB Palette Bypass	
110	8	3/3/2 RGB Palette Index	3/3/2 RGB Palette Bypass
111	8	8 Bit Gray Scale	

4.2.2 Byte 1: Palette Start Address

7	6	5	4	3	2	1	0
RESERVED				START			

POR: All Byte 1 entries power on to 0x00

Bits 7 - 4: RESERVED

Bits 3 - 0: START - Palette Start Address. This field is left shifted six bits and added to the pixel data to form a ten bit palette address for the palette-indexed formats. Refer to Table 10 on page 18 for details on the calculation of the palette address.

4.2.3 Byte 2: Buffer Select

7	6	5	4	3	2	1	0
RESERVED							SEL

POR: All Byte 2 entries power on to 0x00

Bits 7 - 1: RESERVED

Bit 0: SEL - Buffer Select. Chooses buffer A or buffer B for double buffered frame buffer data formats.

0 Buffer A Selected

1 Buffer B Selected

The bits selected are given below:

Bits Per Pixel	WIDCTL Field of Register 0x000A	Buffer A Selected (SEL = 0)	Buffer B Selected (SEL = 1)
4	011	WID Out(07:04)	WID Out(03:00)
4	000 or 100 or 101	Raw Pixel Data(31:28)	Raw Pixel Data(27:24)
8	000 or 100 or 101	Raw Pixel Data(31:24)	Raw Pixel Data(23:16)

4.2.4 Byte 3: Display Control

7	6	5	4	3	2	1	0
RSVD	BOS=0	BVAL					
RSVD	BOS=1	RSVD	XDE	CKE	TRANSP	RSVD	CKSEL

This entry has a dual meaning depending on the value of BOS, bit 6. The BOS bit selects whether the frame buffer and overlay layers are to be blended.

POR: All Byte 3 entries power on to 0x00

Bit 7: RESERVED

Bit 6: BOS - Blend Overlay Select. Enables blending of the frame buffer and the overlay layers.

0 Blend Enabled. Bits 5 - 0 are used for the blend ratio value.

1 Blend Disabled. Bits 5 - 0 are used as miscellaneous control bits.

BOS = 0

Bits 5 - 0: BVAL - Blend Value. The blend value determines the ratio of the blend of the frame buffer and overlay layers. For example, a blend value of 0x3F displays only the overlay layer. A blend value of 0x00 displays only the frame buffer layer. A blend value of 0x1F displays approximately 50% intensity of each layer. When in this blend mode, crosshair display is determined by the crosshair display enable (XDE) bit of byte 3 in the Frame Buffer Window Attribute Table.

BOS = 1

Bit 5: RESERVED

Bit 4: XDE - Crosshair Display Enable. Controls whether the crosshair is visible in the overlay layer.

0 Crosshair display is disabled

1 Crosshair display is enabled

This bit works in conjunction with the Crosshair Control register (register 0x0057) to turn off the crosshair, display a full screen crosshair or limit the crosshair to a window. If crosshair display is enabled by the XDE bit and window ID clipping is enabled, the crosshair is limited to a window.

If the overlay layer is opaque, then the XDE bit in the Overlay Window Attribute Table controls the display of the crosshair. This also applies when the overlay layer is in chroma key mode, even though the overlay layer may be partially or totally transparent. If the overlay layer is transparent, then the XDE bit in the Frame Buffer Window Attribute Table controls the display of the crosshair.

Bit 3: CKE - Chroma Key Enable. Determines if the chroma key function is used.

0 Chroma keying is disabled

1 Chroma keying is enabled

In chroma key mode, the overlay layer is compared to the Chroma Key Register (0 or 1, selected with the CKSEL bit). If the two values match, then the frame buffer layer is displayed. If there is no match, then the overlay layer is displayed. The Chroma Key Mask Register (0 or 1) can be used to limit the bits used in the comparison.

Bit 2: TRANSP - Transparency. Determines whether the frame buffer layer or the overlay layer is displayed.

0 Overlay layer is displayed

1 Frame buffer layer is displayed

If chroma keying is enabled (CKE = 1), the TRANSP bit has no effect.

Bit 1: RESERVED

Bit 0: CKSEL - Chroma Key Select. Determines which of two values to use when calculating if a chroma key match exists.

0 Chroma Key Register 0 (register 0x0020) and Chroma Mask 0 (register 0x0021) are used.

1 Chroma Key Register 1 (register 0x0022) and Chroma Mask 1 (register 0x0023) are used.

5.0 Cursor and Crosshair

5.1 Cursor

The cursor can be either a 32 x 32 or 64 x 64 pixel pattern that is overlaid on the display pixels. The cursor size is selected by a bit in the Cursor Control Register (register 0x004B). When the 32 x 32 pixel cursor is selected, up to four different cursor images can be stored in the internal cursor RAM concurrently.

The cursor may be used in either VGA or non-VGA modes.

5.1.1 Pixel Map Organization

The cursor image is stored in the cursor pixel map. This map is arranged as 1024 x 8 bits. It is updated using index addresses 0x1000 through 0x13FF and read back using index addresses 0x2000 through 0x23FF.

Each pixel of the cursor uses 2 bits, thus 4 cursor pixels are stored in each byte of the array. The entire array is used to contain the 64 x 64 pixel cursor image. Only 256 bytes are required to store a 32 x 32 pixel cursor image. The cursor pixel map is divided into four contiguous slots to allow the storage of four cursor images. The Cursor Control Register (register 0x004B) is used to select one of the four slots for display.

Storage of the cursor within the array starts with the top row. For the 64 x 64 pixel cursor, the first 16 bytes contain the top row, the next 16 bytes the row below the top row, and so on.

For the 32 x 32 pixel cursor, the first 8 bytes contain the top row, the next 8 bytes the row below the top row, and so on. The first image begins at location 0x1000 (0x2000 for reads), the second at location 0x1100 (0x2100 for reads), the third at location 0x1200 (0x2200 for reads), and the fourth at location 0x1300 (0x2300 for reads).

Table 18. 64x64 Cursor Pixel Numbering By Row And Column

	0	**1**	**2**	**3**	**…**	**60**	**61**	**62**	**63**
0	0	1	2	3	…	60	61	62	63
1	64	65	66	67	…	124	125	126	127
2	128	129	130	131	…	188	189	190	191
3	192	193	194	195	…	252	253	254	255
….	…	…	…	…	…	…	…	…	…
60	3840	3841	3842	3843	…	3900	3901	3902	3903
61	3904	3905	3906	3907	…	3964	3965	3966	3967
62	3968	3969	3970	3971	…	4028	4029	4030	4031
63	4032	4033	4034	4035	…	4092	4093	4094	4095

Table 19. Cursor Pixel Mapping

Address	**Data**							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0x2000	Pixel 3		Pixel 2		Pixel 1		Pixel 0	
	Bit 1	Bit 0	Bit 1	Bit 0	Bit 1	Bit 0	Bit 1	Bit 0
0x2001	Pixel 7		Pixel 6		Pixel 5		Pixel 4	
	Bit 1	Bit 0	Bit 1	Bit 0	Bit 1	Bit 0	Bit 1	Bit 0
0x23FE	Pixel 4091		Pixel 4090		Pixel 4089		Pixel 4088	
	Bit 1	Bit 0	Bit 1	Bit 0	Bit 1	Bit 0	Bit 1	Bit 0
0x23FF	Pixel 4095		Pixel 4094		Pixel 4093		Pixel 4092	
	Bit 1	Bit 0	Bit 1	Bit 0	Bit 1	Bit 0	Bit 1	Bit 0

Table 20. Cursor Mode Definitions

Cursor Pixel Map Value	Displayed Cursor Pixel			
	Mode 1	Mode 2	Mode 3	Advanced Cursor Mode
00	Transparent	Color 1	Transparent	Color 0 using Attribute 0
01	Color 1	Color 2	Transparent	Color 1 using Attribute 1
10	Color 2	Transparent	Color 1	Color 2 using Attribute 2
11	Color 3	Highlight	Color 2	Color 3 using Attribute 3

Within a row, the pixels are stored left to right in groups of four. The first byte holds the four leftmost pixels, the second byte the four pixels to the right of the four leftmost, and so on.

Beginning at the least significant portion of a byte, the pixels are stored left to right. The leftmost pixel is stored in bits 1 and 0, the next pixel to the right in bits 3 and 2, and so on.

5.1.2 Processor Access Of The Pixel Map

Writes to and reads from the cursor pixel map are synchronized with the internal video clock. Therefore, the internal video clock must be running for processor accesses to operate properly. If this condition is met, the cursor pixel map may be written to at any time, or read from whenever the cursor display is off—as determined by the MODE field of the Cursor Control Register (register 0x004B).

To begin a read access of the cursor pixel map, the low byte of the index register (I/O port 4) must be loaded before the high byte of the index register (I/O port 5). Loading the high byte of the index register causes a prefetch of the data from the cursor pixel map. This requires that the low byte of the index register have the correct value before the prefetch operation is done.

Subsequent read accesses to sequential locations in the cursor pixel map can be accomplished using the read auto-increment feature. However, when not using auto-increment or when skipping locations, a write to the low byte of the index register must be followed by a write to the high byte of the index register for the proper prefetch to occur (and eventually for the proper cursor pixel map data to be read).

Processor read accesses of the cursor pixel map may disturb the cursor image if it is being displayed at that time. However, no more than one pixel will be disturbed per pixel map access. Processor write accesses of the pixel map will not disturb display of the cursor.

5.1.3 Modes

The cursor operates in one of four modes (Table 20). The first three provide standard operation. The fourth, the advanced cursor mode, adds translucent capability and extended flexibility.

For the advanced cursor mode, the two bit value in the cursor map serves a dual purpose. First, it determines which of four cursor colors to use. Second, it serves as a pointer into the Advanced Function Cursor Attribute Table (register 0x004A). The Advanced Function Cursor Attribute Table contains four, two bit fields. Each field is a code describing how the pixel is to be displayed.

Table 21. Advanced Cursor Attribute Codes

Code	Displayed Cursor Pixel
00	Transparent
01	Opaque
10	Translucent
11	Highlight

Transparent The underlying pixel (normally displayed pixel) is shown. This pixel is the result of the layer blend operation on the frame buffer and overlay layers.

Opaque One of the cursor fill colors from the palette RAM is displayed. The color displayed is determined by the corresponding two bit value in the pixel cursor map.

Translucent Each color band of the underlying pixel (normally displayed pixel) is shifted to the right by one bit. The most significant bit is replaced by a bit from one of the Advanced Function Cursor Color Registers (registers 0x0046 to 0x0049). The particular color register is determined by the corresponding two bit value in the pixel cursor map.

Highlight The most significant bit of each color band of the underlying pixel (normally displayed pixel) is inverted.

5.1.4 Palette Access

The opaque cursor colors are stored in a small extension to the palette RAM. Access to this portion of the palette RAM is identical to that for the main color palette, except for the index address. Four primary and four alternate cursor colors can be loaded into the palette. The alternate colors are used during cursor blinking.

The write addresses for the cursor colors range from index 0x4800 through index 0x4803 for the primary colors and from index 0x4804 through index 0x4807 for the alternate colors.

The read addresses for the cursor colors range from index 0x8800 through index 0x8803 for the primary colors and from index 0x8804 through index 0x8807 for the alternate colors.

Cursor palette data is linearized in the same way and accessed in the same sequence as the main color palette data.

5.1.5 Position

The cursor hot spot is the point within the cursor that is used to locate the cursor's position on the screen. Any pixel within the cursor may be defined as the hot spot.

The Cursor Horizontal Offset Register (register 0x0044) and the Cursor Vertical Offset Register (register 0x0045) hold the unsigned cursor pixel column and row of the hot spot. The range for the column and row offsets is 0 to 31 for the 32 x 32 pixel cursor and 0 to 63 for the 64 x 64 pixel cursor.

The horizontal and vertical position of the cursor hot spot is specified as signed 16 bit two's complement quantities, of which 12 bits of data plus a sign bit are used. These quantities allow positions ranging from -4096 to +4095. The Horizontal and Vertical Position Registers are addressed via indices 0x0040 to 0x0043.

The upper left corner of the screen is located at (0,0). The horizontal coordinate increases from left to right. The vertical coordinate increases from top to bottom. Negative horizontal values are to the left of the display area and negative vertical values are above the display area.

The display of the cursor pattern is clipped by the edges of the screen.

5.1.6 Blinking

The cursor can be made to blink by two different methods. It can blink by switching between the primary and alternate cursor color palettes, or by switching between the primary cursor color palette and transparency. The Cursor Blink Rate Register (register 0x0031) and the Cursor Blink Duty Register (register 0x0032) determine the speed at which the cursor blinks.

At the beginning of the blink cycle, a counter is initialized to one and increments when vertical retrace is detected. The cursor is displayed using the primary color palette until the counter reaches the value in the Blink Duty Register. Then, the alternate palette or transparency is used for the remainder of the cycle. The blink cycle restarts when the counter reaches the value in the Blink Rate Register.

5.1.7 Update

During normal operation, when the cursor control registers at addresses 0x0041 through 0x004B are written, the cursor operation will not be updated until the next vertical retrace.

When the Cursor Horizontal Position Lo register (address 0x0040) is written, updates will be held off completely. The new contents of register 0x0040 and the contents of subsequent writes to registers at addresses 0x0041, 0x0042, and 0x0044 through 0x004B will be held, but no affect on the cursor will be seen. Updates will be resumed, and the new cursor operation will be seen, when the Cursor Vertical Position Hi register (address 0x0043) is written. The first update will occur at the following vertical retrace.

When cursor register updates are being held, reads of the registers will return their previous contents.

The delayed update functions can be overridden by setting the appropriate control bit in the Diagnostics Register (register 0x00FA).

5.2 Crosshair

The crosshair is made up of intersecting horizontal and vertical lines that are overlaid on the display pixels. The width of the horizontal and vertical lines can be 1, 3, 5, or 7 pixels. The width is selected by a field in the Crosshair Control 1 Register (register 0x0057).

The crosshair may be used in either VGA or non-VGA modes.

The crosshair has monochrome and patterned display modes. The monochrome crosshair has programmable width and color. The patterned crosshair has border, fill, and outline patterns, each with programmable width and color.

The crosshair can be clipped to a logical window by use of window ID bits. Window clipping is enabled by setting a bit in the Crosshair Control 1 Register (register 0x0057). The crosshair is then displayed at pixels whose window ID allows crosshair display, as determined by the frame buffer and overlay window attribute tables.

5.2.1 Palette Access

The crosshair colors are stored in a small extension to the palette RAM. Access to this portion of the palette RAM is identical to that for the main color palette, except for the index address. Four primary and four alternate crosshair colors can be loaded into the palette. The alternate colors are used during crosshair blinking.

The write addresses for the crosshair colors range from index 0x4808 through index 0x480B for the primary colors and from index 0x480C through index 0x480F for the alternate colors.

The read addresses for the crosshair colors range from index 0x8808 through index 0x880B for the primary colors and from index 0x880C through index 0x880F for the alternate colors.

Crosshair palette data is linearized in the same way and accessed in the same sequence as the main color palette data.

5.2.2 Position

The horizontal and vertical position of the crosshair center is specified as signed 16 bit two's complement quantities, of which 12 bits of data plus a sign bit are used. These quantities allow positions ranging from -4096 to +4095. The Horizontal and Vertical Position Registers are addressed via indices 0x0050 to 0x0053.

The upper left corner of the screen is located at (0,0). The horizontal coordinate increases from left to right. The vertical coordinate increases from top to bottom. Negative horizontal values are to the left of the display area and negative vertical values are above the display area.

During normal operation, when the position registers at addresses 0x0051 through 0x0053 are written, the crosshair position will not be updated until the next vertical retrace.

When the Crosshair Horizontal Position Lo register (address 0x0050) is written, updates will be held off completely. The new contents of register 0x0050 and the contents of subsequent writes to registers at addresses 0x0051 and 0x0052 will be held, but no affect on the crosshair position will be seen. Updates will be resumed, and the new crosshair position will be seen, when the Crosshair Vertical Position Hi register (address 0x0053) is written. The first update will occur at the following vertical retrace.

When crosshair position register updates are being held, reads of the registers will return their previous contents.

The delayed update function can be overridden by setting the appropriate control bit in the Diagnostics Register (register 0x00FA).

5.2.3 Blinking

Crosshair blinking is controlled by the same registers as those that control cursor blinking.

6.0 VGA

The chip can be placed in VGA mode by properly setting the VGA Control Register (register 0x000B). After reset, the chip is in VGA mode.

In VGA mode, eight bits of pixel data are presented to the palette every load clock cycle. This eight bit bus is connected to input pins on the serial data interface, PIX(159:152). The eight incoming pixel bits are masked by the pixel mask register (processor port 2) before being used to access the palette.

When the chip is in VGA mode, the serializer is disabled and the load clock is used as the video clock. Since there are no individual windows, the crosshair is window clipped, when enabled, to the screen boundaries. The contents of the frame buffer and overlay window attribute tables have no effect on operation while in VGA mode.

Note 1: In non-VGA mode, where the pixel data is coming from VRAM, the incoming load clock (LCLK) is often derived from the outgoing serial clock (SCLK). In VGA mode the operation of SCLK is indeterminate, therefore ***for VGA mode an independent external clock will have to be supplied to the LCLK input.***

Note 2: After reset, the DACs are disabled, and must be explicitly enabled by setting “EN,” the DAC enable bit in the DAC Control Register (register 0x000D). This is a step that is not usually required with other VGA compatible DACs.

7.0 Clocking

There are two on-chip clock generators, the video PLL and the auxiliary PLL. Each PLL clock generator is independently programmable.

The video clock generator provides the fundamental pixel timings when the chip is not operating in VGA mode. It serves generally as the clock both for the internal chip pipelining and the on-card monitor timings.

The auxiliary clock generator provides design convenience and cost saving for the graphics subsystem. The chip makes no internal use of this clock, it simply drives the AUX_PLL_OUT output pin.

7.1 PLL Input

The VID_REF and AUX_REF inputs are reference clocks that the PLL's use in conjunction with programming registers to produce a wide variety of frequencies.

7.2 PLL Output

The supported frequency range for the output of the auxiliary PLL is 8.125 MHz to 100 MHz.

The maximum allowed frequency for the output of the video PLL is 170 MHz / 220 MHz, depending on the product version. The video PLL output is not directly available off-chip. However, two versions of the video clock are provided by the serial clock output and the divided dot clock output.

7.3 PLL Operation and Programming

The two PLLs are generally identical in their operation and programming. A simplified diagram of the PLL is shown in Figure 5. The PLL takes the incoming reference clock, REFCLK, and generates the CLOCKOUT output. The frequency of CLOCKOUT is determined by three programming values contained in registers M, N, and P. A fourth value, C, is required to set the operating points of the analog circuits.

The heart of the PLL is the VCO (voltage controlled oscillator). The VCO can operate over the range of 65 MHz to 170 MHz / 220 MHz (depending on the product speed chosen).

The VCO voltage input value is produced by comparing a divided version of the VCO output with a reference frequency. The value M sets the divide value for the VCO output. Values for M can be 2 through 127 (0 and 1 are illegal). One is added to M, to produce a divide value of 3 through 128.

The internal reference frequency (f_{INTREF}) is produced by dividing the incoming REFCLK, with the divide value set by N. N can range from 0 through 63, and one is added to this value to produce a divider value of 1 through 64.

The divided VCO output frequency is compared to the internal reference, f_{INTREF}, by a phase comparator. The phase comparator drives a charge pump which drives a filter connected to the VCO. A capacitor in the filter develops the voltage supplied to the VCO. The voltage goes up or down depending on whether the phase comparator is sourcing or sinking the charge pump current.

When the divided VCO frequency is equal to f_{INTREF}, no pump current is produced, the voltage to the VCO stays constant, and the VCO frequency stays constant. If the VCO tends to drift in frequency the phase comparator will detect the difference and will drive the filter voltage, via the charge pump, in the appropriate direction. This adjusts the VCO frequency such that the frequency difference at the input to the phase comparator again becomes zero. With a constant f_{INTREF}, the VCO frequency will be locked to the internal reference.

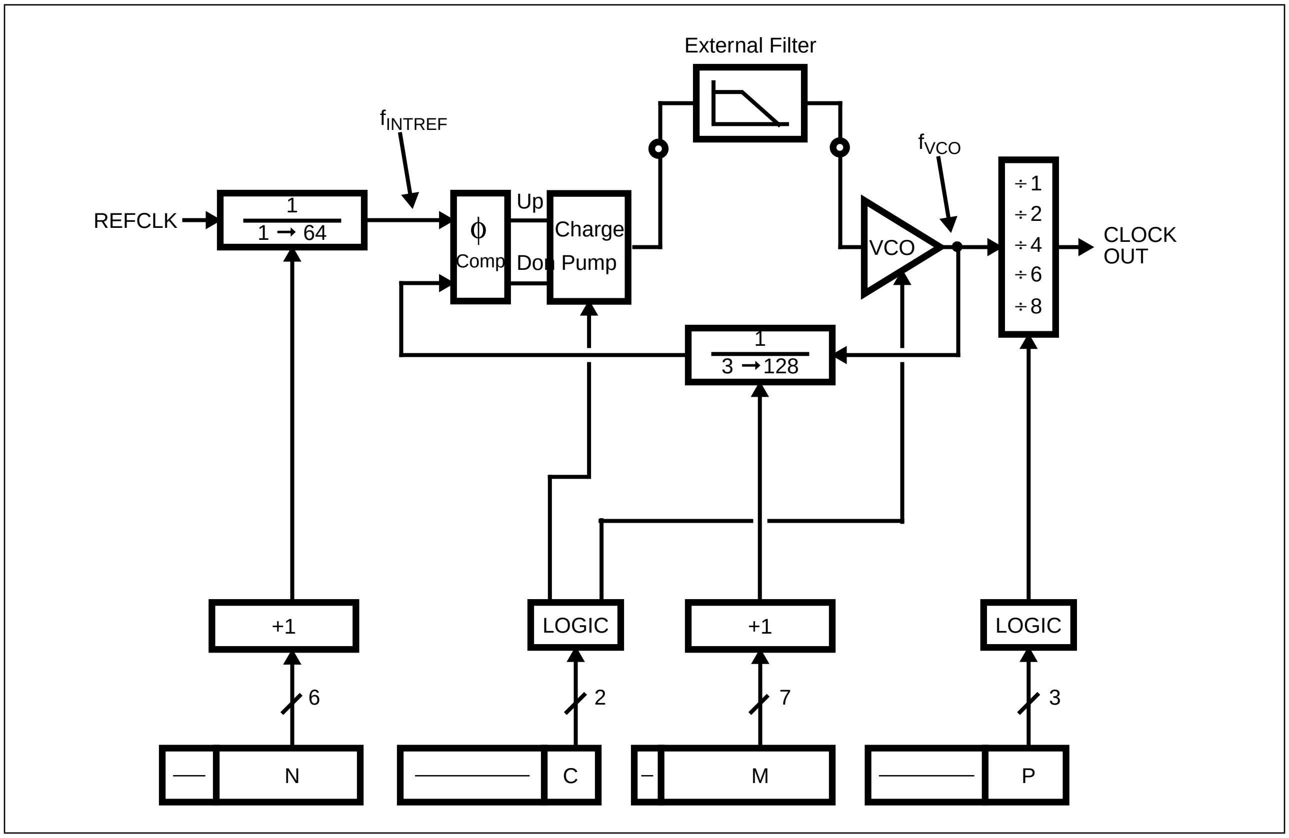


Figure 5. PLL Simplified Diagram

The internal reference frequency is:

(1) $f_{INTREF} = \frac{f_{REFCLK}}{N+1}$

The VCO frequency is:

(2) $f_{VCO} = f_{INTREF} \times (M+1) = f_{REFCLK} \times \frac{M+1}{N+1}$

The output of the VCO is divided down by 1, 2, 4, 6 or 8, depending on the P programming value, to produce the final programmed frequency. For example, a clock frequency of 28 MHz is produced by programming the VCO to oscillate at 4 x 28 = 112 MHz, and then setting P to divide the VCO output by 4.

Valid values for P are 0 through 4 for the video PLL and 1 through 4 for the auxiliary PLL. When P is 0, the divide value is 1; when P = 1, 2, 3 or 4, the divide factor is 2 x P.

The generated CLOCK OUT frequency is:

(3a) $f_{CLOCKOUT} = f_{REFCLK} \times \frac{M+1}{N+1}$; $P = 0$

(3b) $f_{CLOCKOUT} = f_{REFCLK} \times \frac{M+1}{(N+1) \times 2P}$; $P = 1, 2, 3, 4$

Internal to the PLL the charge pump bias current and the VCO gain must be adjusted depending on the frequency of operation. These adjustments are controlled by the C programming bits:

(4a) set $C = 1$; when $65 \text{ MHz} \le f_{VCO} \le 128 \text{ MHz}$

(4b) set $C = 2$; when $f_{VCO} > 128 \text{ MHz}$

Other values of C are reserved.

7.3.1 Additional Constraints

Internal Reference: To avoid excessive jitter, the internal reference frequency, f_{INTREF}, must not be less than 1 MHz. Whenever possible, it is recommended that the internal reference frequency be greater than 2 MHz for best performance.

VCO Frequency Range: The minimum VCO frequency is 65 MHz. The maximum frequency is the maximum speed rating of the product: 170 MHz / 220 MHz. (For example, if it is desired to have a video clock of 150 MHz, it is not permitted to program the PLL to run at 300 MHz with a P value of 1 to divide down to 150 MHz.)

PLL Interaction: The video PLL and the auxiliary PLL will interfere with each other (they will modulate the output frequency of the other PLL) if the higher $f_{CLOCKOUT}$ frequency falls within 3 MHz of an integer multiple of the lower $f_{CLOCKOUT}$ frequency. That is, if f_{higher} is the higher of the two frequencies and f_{lower}, is the lower frequency, then the following equation must be satisfied:

(5) $f_{higher} + 3\text{ MHz} \le n \times f_{lower}$
or $f_{higher} - 3\text{ MHz} \ge n \times f_{lower}$; for each n = 1,2,3,...

7.3.2 Programming Summary

1. Select M, N, and P values that produce the desired frequency using equation (3a) or (3b). Note that for the auxiliary PLL a value of 0 for P is not permitted, so equation (3a) is not valid for this PLL.
2. Verify that f_{INTREF} is not less than 1 MHz, using equation (1).
3. Verify that f_{VCO} is not less than 65 MHz and not greater than 170 MHz / 220 MHz, using equation (2).
4. Verify that the higher $f_{CLOCKOUT}$ frequency of the 2 PLLs does not fall within 3 MHz of an integral multiple of the lower $f_{CLOCKOUT}$ frequency (equation (5)).
5. If all of the above conditions are met then M, N, and P are valid values. Select the appropriate C value using equation (4a) or (4b).

7.3.3 Glitching on Frequency Change

When the operating frequency of either PLL is changed by changing one of the programming register values, the transition from the original frequency to the new frequency can either occur smoothly or can glitch, depending upon the following:

1. If the P bits are not changed, then changing the M and N bits will not cause a glitch.
2. If the P value is changed, then the PLL output can glitch. For the video PLL there is no protection against this situation.

   For the auxiliary PLL there is additional logic (not shown) which causes the 2,4,6,8 VCO divider circuit to update synchronously with the VCO output when the P value is changed, such that changing P will not cause AUX_PLL_OUT to glitch. (See 10.21 “0x0014 Auxiliary PLL Reference Divide” on page 61 for a description of the register update sequence required to engage this logic.)
3. Caution should be used when changing the C value. Although the output will not glitch, the frequency can change rapidly to intermediate values before settling to the new value.

7.3.4 PLL Reset

During reset, the auxiliary PLL is bypassed and the reference clock is driven onto the output pin, AUX_PLL_OUT. Following reset, the auxiliary PLL programming registers are initialized so that the auxiliary PLL is no longer bypassed. The auxiliary PLL uses the reference clock to form an output with the same frequency as the reference. The output auxiliary clock is guaranteed to transition without glitches at the completion of the reset cycle.

During reset, the video PLL is disabled. Following the reset cycle, the video PLL programming registers must be written to the values required for the frequency desired.

7.3.5 PLL Disable

Each PLL can be separately disabled. In the disabled state, the PLL does not respond to the M and N programming values. This causes the internal VCO to oscillate somewhere in the range of 20 KHz to 1 MHz.

Changing the C value may change the VCO frequency, but the frequency will stay within the 20 KHz to 1 MHz range.

The VCO output continues to be affected by the P value, so the actual PLL output will be 20 KHz through 1 MHz divided by 1, 2, 4, 6 or 8, for the video PLL, and 20 KHz through 1 MHz divided by 2, 4, 6 or 8, for the auxiliary PLL.

The video PLL is disabled by setting the EN bit in Video PLL Control Register (register 0x0013) to zero. The auxiliary PLL is disabled by setting the EN bit in the Auxiliary PLL Control Register (register 0x0017) to zero.

7.4 Video Clock

The video clock, or dot clock, is the internal clock used to propagate pixel data through the chip. The video clock must be running to access the palettes and the cursor pixel map. The maximum frequency of this clock is 170 MHz / 220 MHz depending on the product rated speed.

The video clock has three possible sources. When the chip is in VGA mode, the source of the video clock is the load clock. Otherwise, the video clock is supplied by the programmable video PLL or an external ECL differential receiver pair. The choice between the two is register selectable.

7.5 Serial Clock

The serial clock (SCLK) is intended for clocking of the serial outputs of the VRAMs. The serial clock is formed by dividing down the video clock. The divide factor is dependent on the serializer multiplex mode. For example, when using 4:1 multiplex mode, the serial clock frequency is one quarter the frequency of the video clock.

The packed 16:3 mode is a special case. Three serial clocks are produced for every 16 video clocks.

Note: In VGA mode the operation of SCLK is indeterminate.

7.6 Load Clock

The load clock (LCLK) is an input clock that is used to latch the pixel data and monitor control signals (sync and blank) into the chip. In non-VGA mode it is derived from the serial clock output of the chip. When the chip is in VGA mode, the load clock is used as the video clock to advance the data through the internal chip pipeline.

Note: In non-VGA mode, the RGB640 monitors the transitions of LCLK relative to the internal SCLK. If the RGB640 determines that there is too much skew between LCLK and SCLK, it will invert the outgoing SCLK. This ensures that the signals latched by LCLK will be reliably transferred to latches clocked by the internal SCLK.

Therefore, in non-VGA mode LCLK must be derived from SCLK.

7.7 Divided Dot Clock

The divided dot clock is simply the video clock divided by 1, 2, 4, or 8. The maximum supported output frequency of the divided dot clock is 100 MHz, so some divide values are not valid when the video clock is operating at higher frequencies. The divided dot clock output is provided for general card use and is not used internally. Although the divided dot clock and the serial clock are both derived from the video clock, there is otherwise no relationship between the two clocks. In particular, the two clocks have no associated phase relationship.

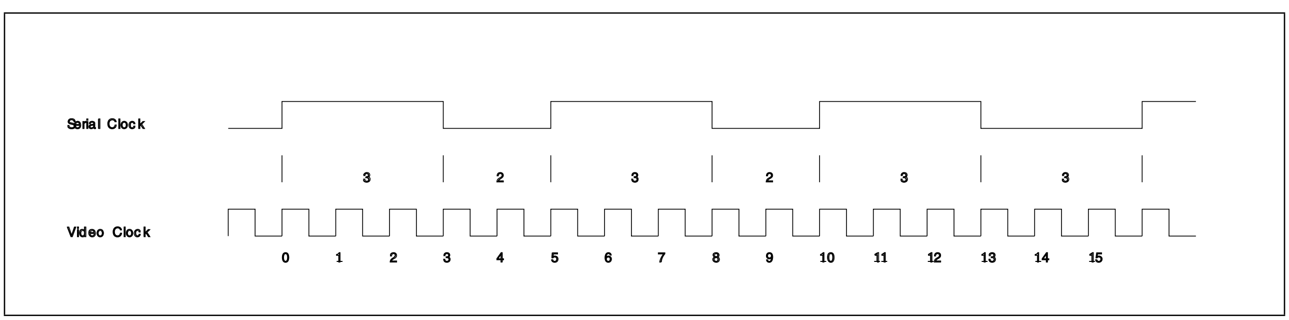


Figure 6. Packed Mode Serial Clock

8.0 Diagnostics

8.1 MISR

A Multiple Input Shift Register (MISR) is used to enhance testability of the frame buffer to RGB640 interface. The MISR continually processes the 30 bit digital DAC pixel input for a frame of data into a signature and stores it in 4 eight bit registers (MISR Signature Registers, registers 0x00FC to 0x00FF).

Note: the contents of the MISR Signature Registers are the ***inverse*** of the calculated signature.

This signature can then be read from the product and compared with the correct signature for the specific frame of data used to determine if a fault exists. By using the VRAM Mask Registers (registers 0x00F0 to 0x00F2) to block specific pixel inputs and by using MISR diagnostics, the fault can be isolated to a section of circuitry, card net, or VRAM module.

The MISR is enabled by setting the EN bit in the MISR Control / Status Register (register 0x00FB). After enabling, the MISR is reset to 0x3FFFFFFF at the next active vertical blanking time and begins accumulating a frame signature when vertical blanking becomes inactive.

The MISR enable bit is not reset automatically. To collect another signature, the MISR must be disabled for at least 1 frame, then enabled again. It is recommended that the monitor be blanked while the MISR operation runs test frames by disabling the DAC outputs.

8.1.1 MISR Algorithm

The MISR signature is generated using the 30 bit DAC pixel input and implementing the polynomial $x^{23} + x^2 + x^1 + x^0$. A continuous XOR of data from the output of the cursor blend logic with shifted MISR register data and feedback of the 29th register bit into selected positions, generates the signature.

8.1.2 MISR Example

In the example below, the first row represents the new input pixel data, the second row is the previous MISR data shifted 1 bit to the left, the third row is the 29th bit of data fed back into selected bit positions and the fourth row is the result of the 2 or 3 input XOR operation.

The input data used in this example has a value of 0x00000200 for each pixel clock cycle, the MISR register reset value is 0x3FFFFFFF and the signature is compiled for 2 pixel cycles. MISR bit 0 (least significant) is on the right, bit 29 (most significant) on the left.

```
00 0000 0000 0000 0000 0010 0000 0000 Pixel data
11 1111 1111 1111 1111 1111 1111 111  Shifted MISR
        1                         111 Bit 29
_____________________________________
11 1111 0111 1111 1111 1101 1111 1001 Cycle 1
                                      0x3F7FFDF9


00 0000 0000 0000 0000 0010 0000 0000 Pixel data
11 1110 1111 1111 1111 1011 1111 001  Shifted MISR
        1                         111 Bit 29
_____________________________________
11 1110 0111 1111 1111 1001 1111 0101 Cycle 2
                                      0x3E7FF9F5
```

A third cycle would yield the value 0x3C7FF1ED.

8.1.3 VRAM Mask Registers

The VRAM Mask Registers (registers 0x00F0 to 0x00F2) can be used to isolate defective VRAM modules, card nets, or chip circuitry. If an incorrect MISR frame signature is obtained with masking disabled, additional fault isolation can be achieved by collecting new signatures with selected VRAM inputs masked to zero and comparing them with correct signatures.

8.2 Diagnostics Register

The Diagnostics Register (register 0x00FA) has a single bit that controls update of the window attribute tables. During normal operation, writing to a window attribute table will not update the table until the next vertical retrace following the resetting of the WATCTL bit in the Update Control Register (register 0x000E). Read back of the window attribute table will return old data until then.

During diagnostic operation, the contents of the window attribute table are updated immediately.

9.0 Software Interface

9.1 Processor I/O Space

The RGB640 supports eight processor I/O ports, one of which is reserved. These ports are selectable via three register select pins, RS(02:00). Two indirect schemes are used to access all of the internal registers and arrays.

The first four ports are provided for standard VGA compatibility in accessing the palette. Of these four ports, only one port is used to directly access a register, the Pixel Mask Register. The other three ports are used to read and write a 256 x 8 subsection of the palette.

The second four ports are used for general access of all internal registers and arrays using an index scheme for extending the address. Two of the ports are used to load high and low bytes of an index register. A third port is used to read data from or write data to a register or array location specified by the index address. The fourth port in this group is reserved.

All ports are readable and writable.

Table 22. I/O Port Map

RS(02:00)	Operation
000	Palette Address (write mode)
001	Palette Data
010	Pixel Mask (direct access)
011	Palette Address (read mode)
100	Index(07:00)
101	Index(15:08)
110	Index Register Data Port
111	Reserved

Port 0 Palette address (write mode). Writing this port initializes the logic for write operations to the palette. Subsequent writes to the palette data port will load internal palette color registers and cause these register contents to be written into the palette.

Port 1 Palette data. This port is used to read from and write to a 256 x 8 section of the palette. To write to the palette, port 0 must be written with the starting palette address before writing to the palette data port. To read from the palette, the port 3 must be written with the starting palette address before reading from the palette data port.

The palette data port can be accessed as either six bits or eight bits wide. This is determined by the PSIZE bit in the VGA Control Register (register 0x000B). Three accesses to the palette data port are required to complete one read or one write of the palette. Each transfer accesses another color band in order from red to green to blue.

Port 2 Palette mask. This port accesses a register whose contents mask the VGA pixel data inputs. Access to this port is asynchronous to palette access, and will not disturb the palette access sequence. Temporary color disturbances can be expected if the mask is changed while displaying pixels through the palette.

Port 3 Palette address (read mode). Writing this port initializes the logic for read operations from the palette. Data from the palette is loaded into internal color palette registers. Subsequent reads from the palette data port will read these color palette registers.

Reading port 3 returns either the current palette address (read mode) or the status of the last palette access. The data returned by a read of this port is determined by the RDBK bit of the VGA Control Register (register 0x000B). If this bit is set so that a read of this port will return status, a value of 0x00 is returned if the last write to the palette address was for write mode (port 0). A value of 0x03 is returned if the last write to the palette address was for read mode (port 3).

Port 4 Index lo. This port provides access to the least significant 8 bits of the 16 bit index register. The index register provides access to all internal register and array locations in conjunction with the index data port.

Port 5 Index hi. This port provides access to the most significant 8 bits of the 16 bit index register. The index register

	provides access to all internal register and array locations in conjunction with the index data port.
Port 6	Index data. A read from or write to this port performs that operation to the register or array location addressed by the index register.

9.2 VGA Palette Access

The palette can be accessed in two ways. For VGA compatibility, processor I/O ports 0, 1, and 3 should be used. For VGA accesses, only the first 256 locations of the palette can be accessed. If auto-increment is used, the address will wrap back to 0x00 after reaching 0xFF.

Non-VGA type access is provided through processor I/O ports 4, 5, and 6. All 1024 palette locations are accessible using non-VGA type accesses.

9.2.1 Palette Write

VGA palette writes must be begun by writing the palette address (write mode) register. This provides a starting address for writes and initializes the internal write logic.

Palette writes are then performed by writing to the palette data port three times. The first write is used to load the red palette, the second to load the green palette, and the third to load the blue palette. Following the third write, the palette address register will be incremented. Thus, continuous writes to the palette data port will load the palette, stepping through the palette addresses in ascending order.

9.2.2 Palette Read

VGA palette reads must be begun by writing the palette address (read mode) register. This provides a starting address for reads and initializes the internal read logic.

Palette reads are then performed by reading from the palette data port three times. The first read returns the value in the red palette, the second returns the value in the green palette, and the third returns the value in the blue palette. Following the third read, the palette address register will be incremented. Thus, continuous reads from the palette data port will read the palette, stepping through the palette addresses in ascending order.

9.2.3 6 / 8 Bit Access

VGA reads and writes to the palette can be either 6 bits or 8 bits wide per color palette. The mode of operation is determined by the PSIZE bit in the VGA Control Register (register 0x000B). When writing to the palette, the incoming data is linearized before being written into the palette. This is done by shifting the incoming data to the most significant portion of a ten bit value and then replacing any missing least significant bits in the ten bit value with the most significant bits of the incoming data. See Tables 23, 24 and 25 on page 43.

When the VGA palette is accessed as 6 bits per band, those six bits occupy the least significant 6 bits on the processor bus, DATA(07:00). During writes, bits DATA(07:06) are discarded. During reads, bits DATA(07:06) return zeroes.

9.3 Indexed Access

To access one of the internal registers or array locations, the index must be set to the desired location. Table 29 on page 44 contains a summary of all internal registers and array location addresses. The index(15:00) register is set up by writing to processor I/O ports 4 and 5. The index data port is then used to access the desired internal register or array location.

The index(15:00) register can auto-increment after each access to the index data port. The RDAI and WRAI bits of the Update Control Register (register 0x000E) are used to enable or disable the auto-increment function.

9.4 Palette Access

The palette has 1024 locations and supports ten bit color. There are an additional 16 palette locations used for cursor and crosshair colors. The palette is accessed as an eight or ten bit palette using indexed access and processor I/O port 6.

When accessing the palette, multiple data cycles are required to read from or write to each palette location. For eight bit modes, three separate byte transfers are required for each location. For ten bit modes, data is accessed with a choice of four or six separate byte transfers for each location. These options are determined by the bit settings in the Miscellaneous Configuration Register (register 0x000A).

9.4.1 8 Bit Palette

In this mode, the palette is accessed as a 1024 x 24 bit array. Three transfers are required to access each palette location (see Table 26 on page 43). The read or write logic is initialized by writing to the index hi port (port 5). The address that is written to the index register prepares the logic for either a read or write sequence through the data port.

9.4.1.1 Writing The Palette

To write to a palette location, the index register must be loaded with the address of the palette. The address is equal to 0x4000 plus the location offset into the palette. Subsequent writes to the index data port (port 6) will update the palette after each third access.

The eight bit data is linearized before being written to the palette. After the third write, the index register will auto-increment to the next palette location provided that this function has been enabled by the Update Control Register (register 0x000E).

9.4.1.2 Reading The Palette

To begin a read access of the palette, the low byte of the index register (I/O port 4) must be loaded before the high byte of the index register (I/O port 5). Loading the high byte of the index register causes a prefetch of the data from the palette. This requires that the low byte of the index register have the correct value before the prefetch operation is done.

Subsequent read accesses to sequential locations in the palette can be accomplished using the read auto-increment feature. However, when not using auto-increment or when skipping locations, a write to the low byte of the index register must be followed by a write to the high byte of the index register for the proper prefetch to occur (and eventually for the proper palette data to be read).

To read from a palette location, the index register must be loaded with the address of the palette. The address is equal to 0x8000 plus the location offset into the palette. Subsequent reads from the index data port (port 6) will present data from the red, green, and blue palettes in that order.

After the third read, the index register will auto-increment to the next palette location provided that this function has been enabled by the Update Control Register (register 0x000E).

9.4.2 10 Bit Palette

In this mode, the palette is accessed as a 1024 x 30 bit array. Four or six transfers are required to access each palette location. The read or write logic is initialized by writing to the index hi port (port 5). The address that is written to the index register prepares the logic for either a read or write sequence through the data port.

9.4.2.1 Palette Sequences

The two sequences available for 10 bit palette access are shown in Table 27 and Table 28 on page 43. A six byte sequence keeps the color bands together, while the four byte sequence writes all of the least significant bits from all three color bands in the fourth transfer.

Reserved bits in the sequence are discarded during writes, and return zeroes during reads.

9.5 Cursor Pixel Map Access

The cursor pixel map can contain a 64 x 64 array of two bit pixels. For access from the processor port, the pixel map is organized as a 256 x 8 array. More information about the organization of the cursor pixel map is contained in section 5.1.1, “Pixel Map Organization” on page 29.

The read and write address spaces for the pixel map are separated, just as they are for the palette. Writes to the pixel map begin at index 0x1000. Reads from the pixel map begin at index 0x2000.

Table 23. Red 6 / 8 Bit Linearization

	Red(09)	Red(08)	Red(07)	Red(06)	Red(05)	Red(04)	Red(03)	Red(02)	Red(01)	Red(00)
6 Bit	DATA(05)	DATA(04)	DATA(03)	DATA(02)	DATA(01)	DATA(00)	DATA(05)	DATA(04)	DATA(03)	DATA(02)
8 Bit	DATA(07)	DATA(06)	DATA(05)	DATA(04)	DATA(03)	DATA(02)	DATA(01)	DATA(00)	DATA(07)	DATA(06)

Table 24. Green 6 / 8 Bit Linearization

	Green(09)	Green(08)	Green(07)	Green(06)	Green(05)	Green(04)	Green(03)	Green(02)	Green(01)	Green(00)
6 Bit	DATA(05)	DATA(04)	DATA(03)	DATA(02)	DATA(01)	DATA(00)	DATA(05)	DATA(04)	DATA(03)	DATA(02)
8 Bit	DATA(07)	DATA(06)	DATA(05)	DATA(04)	DATA(03)	DATA(02)	DATA(01)	DATA(00)	DATA(07)	DATA(06)

Table 25. Blue 6 / 8 Bit Linearization

	Blue(09)	Blue(08)	Blue(07)	Blue(06)	Blue(05)	Blue(04)	Blue(03)	Blue(02)	Blue(01)	Blue(00)
6 Bit	DATA(05)	DATA(04)	DATA(03)	DATA(02)	DATA(01)	DATA(00)	DATA(05)	DATA(04)	DATA(03)	DATA(02)
8 Bit	DATA(07)	DATA(06)	DATA(05)	DATA(04)	DATA(03)	DATA(02)	DATA(01)	DATA(00)	DATA(07)	DATA(06)

Table 26. Three Byte Access Sequence

Access	DATA(07)	DATA(06)	DATA(05)	DATA(04)	DATA(03)	DATA(02)	DATA(01)	DATA(00)
1	Red(07)	Red(06)	Red(05)	Red(04)	Red(03)	Red(02)	Red(01)	Red(00)
2	Green(07)	Green(06)	Green(05)	Green(04)	Green(03)	Green(02)	Green(01)	Green(00)
3	Blue(07)	Blue(06)	Blue(05)	Blue(04)	Blue(03)	Blue(02)	Blue(01)	Blue(00)

Table 27. Four Byte Access Sequence

Access	DATA(07)	DATA(06)	DATA(05)	DATA(04)	DATA(03)	DATA(02)	DATA(01)	DATA(00)
1	Red(09)	Red(08)	Red(07)	Red(06)	Red(05)	Red(04)	Red(03)	Red(02)
2	Green(09)	Green(08)	Green(07)	Green(06)	Green(05)	Green(04)	Green(03)	Green(02)
3	Blue(09)	Blue(08)	Blue(07)	Blue(06)	Blue(05)	Blue(04)	Blue(03)	Blue(02)
4	reserved	reserved	Red(01)	Red(00)	Green(01)	Green(00)	Blue(01)	Blue(00)

Table 28. Six Byte Access Sequence

Access	DATA(07)	DATA(06)	DATA(05)	DATA(04)	DATA(03)	DATA(02)	DATA(01)	DATA(00)
1	Red(09)	Red(08)	Red(07)	Red(06)	Red(05)	Red(04)	Red(03)	Red(02)
2	Red(01)	Red(00)	reserved	reserved	reserved	reserved	reserved	reserved
3	Green(09)	Green(08)	Green(07)	Green(06)	Green(05)	Green(04)	Green(03)	Green(02)
4	Green(01)	Green(00)	reserved	reserved	reserved	reserved	reserved	reserved
5	Blue(09)	Blue(08)	Blue(07)	Blue(06)	Blue(05)	Blue(04)	Blue(03)	Blue(02)
6	Blue(01)	Blue(00)	reserved	reserved	reserved	reserved	reserved	reserved

9.6 Indexed Address Space

For correct operation and to preserve upward compatibility, the registers listed as reserved must not be written to. Within a register, individual bits listed as reserved must be set to zeroes.

Table 29. Single Cycle Addresses

Index (15:00)	Register Addressed	Page
0x0000	Identification	48
0x0001	Revision Level	48
0x0002	Serializer Raw Pixel (07:00) Control	49
0x0003	Serializer Raw Pixel (15:08) Control	49
0x0004	Serializer Raw Pixel (23:16) Control	49
0x0005	Serializer Raw Pixel (31:24) Control	50
0x0006	Serializer WID Out (03:00) Control	50
0x0007	Serializer WID Out (07:04) Control	50
0x0008	Serializer Mode	51
0x0009	Pixel Interleave	51
0x000A	Miscellaneous Configuration	52
0x000B	VGA Control	53
0x000C	DAC Compare / Monitor ID	54
0x000D	DAC Control	55
0x000E	Update Control	56
0x000F	Sync Control	57
0x0010	Video PLL Reference Divide	58
0x0011	Video PLL Multiplier	58
0x0012	Video PLL Output Divide	59
0x0013	Video PLL Control	60
0x0014	Auxiliary PLL Reference Divide	61
0x0015	Auxiliary PLL Multiplier	61
0x0016	Auxiliary PLL Output Divide	62
0x0017	Auxiliary PLL Control	63
0x0018 - 0x001F	Reserved	
0x0020	Chroma Key 0	64
0x0021	Chroma Key Mask 0	64
0x0022	Chroma Key 1	64
0x0023	Chroma Key Mask 1	64
0x0024 - 0x002F	Reserved	
0x0030	Cursor / Crosshair Control	65
0x0031	Cursor Blink Rate	65
0x0032	Cursor Blink Duty Cycle	65
0x0033 - 0x003F	Reserved	
0x0040	Cursor Horizontal Position Lo	66
0x0041	Cursor Horizontal Position Hi	66
0x0042	Cursor Vertical Position Lo	67
0x0043	Cursor Vertical Position Hi	67

Table 29. Single Cycle Addresses (Continued)

Index (15:00)	Register Addressed	Page
0x0044	Cursor Horizontal Offset	68
0x0045	Cursor Vertical Offset	68
0x0046	Advanced Function Cursor Color 0	69
0x0047	Advanced Function Cursor Color 1	69
0x0048	Advanced Function Cursor Color 2	70
0x0049	Advanced Function Cursor Color 3	70
0x004A	Advanced Function Cursor Attribute Table	71
0x004B	Cursor Control	72
0x004C - 0x004F	Reserved	
0x0050	Crosshair Horizontal Position Lo	73
0x0051	Crosshair Horizontal Position Hi	73
0x0052	Crosshair Vertical Position Lo	74
0x0053	Crosshair Vertical Position Hi	74
0x0054	Crosshair Pattern Color	75
0x0055	Crosshair Vertical Pattern	76
0x0056	Crosshair Horizontal Pattern	76
0x0057	Crosshair Control 1	77
0x0058	Crosshair Control 2	78
0x0059 - 0x006F	Reserved	
0x0070	YUV Conversion Coefficient K1	79
0x0071	YUV Conversion Coefficient K2	79
0x0072	YUV Conversion Coefficient K3	79
0x0073	YUV Conversion Coefficient K4	79
0x0074 - 0x00EF	Reserved	
0x00F0	VRAM Mask Register 0	80
0x00F1	VRAM Mask Register 1	80
0x00F2	VRAM Mask Register 2	81
0x00F3 - 0x00F9	Reserved	
0x00FA	Diagnostics	82
0x00FB	MISR Control / Status	82
0x00FC	MISR Signature 0	83
0x00FD	MISR Signature 1	83
0x00FE	MISR Signature 2	83
0x00FF	MISR Signature 3	83
0x0100 - 0x013F	Frame Buffer WAT	47
0x0140 - 0x01FF	Reserved	
0x0200 - 0x023F	Overlay WAT	47
0x0240 - 0x0FFF	Reserved	
0x1000 - 0x13FF	Cursor Pixel Map Write	29, 42
0x1400 - 0x1FFF	Reserved	
0x2000 - 0x23FF	Cursor Pixel Map Read	29, 42
0x2400 - 0x3FFF	Reserved	

Table 30. Multiple Cycle Addresses

Index(15:00)	Register Accessed	Page
0x4000 - 0x43FF	Main Color Palette Write	42
0x4400 - 0x47FF	Reserved	
0x4800	Cursor Color 0 Write	31
0x4801	Cursor Color 1 Write	31
0x4802	Cursor Color 2 Write	31
0x4803	Cursor Color 3 Write	31
0x4804	Alternate Cursor Color 0 Write	31
0x4805	Alternate Cursor Color 1 Write	31
0x4806	Alternate Cursor Color 2 Write	31
0x4807	Alternate Cursor Color 3 Write	31
0x4808	Crosshair Color 0 Write	32
0x4809	Crosshair Color 1 Write	32
0x480A	Crosshair Color 2 Write	32
0x480B	Crosshair Color 3 Write	32
0x480C	Alternate Crosshair Color 0 Write	32
0x480D	Alternate Crosshair Color 1 Write	32
0x480E	Alternate Crosshair Color 2 Write	32
0x480F	Alternate Crosshair Color 3 Write	32
0x4810 - 0x4FFF	Reserved	
0x5000 - 0x7FFF	Reserved, read 3-states drivers	
0x8000 - 0x83FF	Main Color Palette Read	42
0x8400 - 0x87FF	Reserved	
0x8800	Cursor Color 0 Read	31
0x8801	Cursor Color 1 Read	31
0x8802	Cursor Color 2 Read	31
0x8803	Cursor Color 3 Read	31
0x8804	Alternate Cursor Color 0 Read	31
0x8805	Alternate Cursor Color 1 Read	31
0x8806	Alternate Cursor Color 2 Read	31
0x8807	Alternate Cursor Color 3 Read	31
0x8808	Crosshair Color 0 Read	32
0x8809	Crosshair Color 1 Read	32
0x880A	Crosshair Color 2 Read	32
0x880B	Crosshair Color 3 Read	32
0x880C	Alternate Crosshair Color 0 Read	32
0x880D	Alternate Crosshair Color 1 Read	32
0x880E	Alternate Crosshair Color 2 Read	32
0x880F	Alternate Crosshair Color 3 Read	32
0x8810 - 0x8FFF	Reserved	
0x9000 - 0x9FFF	Reserved, read 3-states drivers	

9.7 Frame Buffer WAT Index Assignments

Table 31. Frame Buffer WAT Addresses

Window	Pixel Type	Palette Start	Buffer A / B	Control
0	0x0100	0x0101	0x0102	0x0103
1	0x0104	0x0105	0x0106	0x0107
2	0x0108	0x0109	0x010A	0x010B
3	0x010C	0x010D	0x010E	0x010F
4	0x0110	0x0111	0x0112	0x0113
5	0x0114	0x0115	0x0116	0x0117
6	0x0118	0x0119	0x011A	0x011B
7	0x011C	0x011D	0x011E	0x011F
8	0x0120	0x0121	0x0122	0x0123
9	0x0124	0x0125	0x0126	0x0127
10	0x0128	0x0129	0x012A	0x012B
11	0x012C	0x012D	0x012E	0x012F
12	0x0130	0x0131	0x0132	0x0133
13	0x0134	0x0135	0x0136	0x0137
14	0x0138	0x0139	0x013A	0x013B
15	0x013C	0x013D	0x013E	0x013

9.8 Overlay WAT Index Assignments

Table 32. Overlay WAT Addresses

Window	Pixel Type	Palette Start	Buffer A / B	Control Bits
0	0x0200	0x0201	0x0202	0x0203
1	0x0204	0x0205	0x0206	0x0207
2	0x0208	0x0209	0x020A	0x020B
3	0x020C	0x020D	0x020E	0x020F
4	0x0210	0x0211	0x0212	0x0213
5	0x0214	0x0215	0x0216	0x0217
6	0x0218	0x0219	0x021A	0x021B
7	0x021C	0x021D	0x021E	0x021F
8	0x0220	0x0221	0x0222	0x0223
9	0x0224	0x0225	0x0226	0x0227
10	0x0228	0x0229	0x022A	0x022B
11	0x022C	0x022D	0x022E	0x022F
12	0x0230	0x0231	0x0232	0x0233
13	0x0234	0x0235	0x0236	0x0237
14	0x0238	0x0239	0x023A	0x023B
15	0x023C	0x023D	0x023E	0x023F

10.0 Register Descriptions

10.1 0x0000 Identification

7	6	5	4	3	2	1	0
IDLO							

POR: 0x1C

Bits 7 - 0: IDLO - The low byte of the product identification code. (0x1C)

10.2 0x0001 Identification / Revision Level

7	6	5	4	3	2	1	0
REV				IDHI			

POR: 0x12

Bits 7 - 4: REV - The product revision level code. (0x1)

Bits 3 - 0: IDHI - The high nibble of the product identification code. (0x2)

10.3 0x0002 Serializer Raw Pixel(07:00) Control

7	6	5	4	3	2	1	0
RESERVED		SKIP		START			

POR: 0x00

Bits 7 - 6: RESERVED

Bits 5 - 4: SKIP - determines the number of bytes on the serial data interface, PIX(127:000), that are skipped over from the location of the corresponding byte in the first pixel, Raw Pixel Data(07:00), to find this byte in the second pixel. The skip value is applied again in the same way to find this byte in the third pixel, and so on. A skip value of zero yields consecutive bytes from the serial data interface.

Bits 3 - 0: START - determines which byte on the serial data interface, PIX(127:000), is used to form this byte of the first serialized pixel, Raw Pixel Data(07:00). Refer to Table 1 on page 6 for the mapping of START values to bytes on the serial data interface.

10.4 0x0003 Serializer Raw Pixel(15:08) Control

7	6	5	4	3	2	1	0
RESERVED		SKIP		START			

POR: 0x00

Bits 7 - 6: RESERVED

Bits 5 - 4: SKIP - determines the number of bytes on the serial data interface, PIX(127:000), that are skipped over from the location of the corresponding byte in the first pixel, Raw Pixel Data(15:08), to find this byte in the second pixel. The skip value is applied again in the same way to find this byte in the third pixel, and so on. A skip value of zero yields consecutive bytes from the serial data interface.

Bits 3 - 0: START - determines which byte on the serial data interface, PIX(127:000), is used to form this byte of the first serialized pixel, Raw Pixel Data(15:08). Refer to Table 1 on page 6 for the mapping of START values to bytes on the serial data interface.

10.5 0x0004 Serializer Raw Pixel(23:16) Control

7	6	5	4	3	2	1	0
RESERVED		SKIP		START			

POR: 0x00

Bits 7 - 6: RESERVED

Bits 5 - 4: SKIP - determines the number of bytes on the serial data interface, PIX(127:000), that are skipped over from the location of the corresponding byte in the first pixel, Raw Pixel Data(23:16), to find this byte in the second pixel. The skip value is applied again in the same way to find this byte in the third pixel, and so on. A skip value of zero yields consecutive bytes from the serial data interface.

Bits 3- 0: START - determines which byte on the serial data interface, PIX(127:000), is used to form this byte of the first serialized pixel, Raw Pixel Data(23:16). Refer to Table 1 on page 6 for the mapping of START values to bytes on the serial data interface.

10.6 0x0005 Serializer Raw Pixel(31:24) Control

7	6	5	4	3	2	1	0
RESERVED		SKIP		START			

POR: 0x00

Bits 7 - 6: RESERVED

Bits 5 - 4: SKIP - determines the number of bytes on the serial data interface, PIX(127:000), that are skipped over from the location of the corresponding byte in the first pixel, Raw Pixel Data(31:24), to find this byte in the second pixel. The skip value is applied again in the same way to find this byte in the third pixel, and so on. A skip value of zero yields consecutive bytes from the serial data interface.

Bits 3- 0: START - determines which byte on the serial data interface, PIX(127:000), is used to form this byte of the first serialized pixel, Raw Pixel Data(31:24). Refer to Table 1 on page 6 for the mapping of START values to bytes on the serial data interface.

10.7 0x0006 Serializer WID Out(03:00) Control

7	6	5	4	3	2	1	0
RESERVED		SKIP		START			

POR: 0x00

Bits 7 - 6: RESERVED

Bits 5 - 4: SKIP - determines the number of nibbles on the serial data interface, PIX(159:120), that are skipped over from the location of the corresponding nibble in the first pixel, WID Out(03:00), to find this nibble in the second pixel. The skip value is applied again in the same way to find this nibble in the third pixel, and so on. A skip value of zero yields consecutive nibbles from the serial data interface.

Bits 3- 0: START - determines which nibble on the serial data interface, PIX(159:120), is used to form this nibble of the first serialized pixel, WID Out(03:00). Refer to Table 2 on page 8 for the mapping of START values to nibbles on the serial data interface.

10.8 0x0007 Serializer WID Out(07:04) Control

7	6	5	4	3	2	1	0
RESERVED		SKIP		START			

POR: 0x00

Bits 7 - 6: RESERVED

Bits 5 - 4: SKIP - determines the number of nibbles on the serial data interface, PIX(159:120), that are skipped over from the location of the corresponding nibble in the first pixel, WID Out(07:04), to find this nibble in the second pixel. The skip value is applied again in the same way to find this nibble in the third pixel, and so on. A skip value of zero yields consecutive nibbles from the serial data interface.

its 3- 0: START - determines which nibble on the serial data interface, PIX(159:120), is used to form this nibble of the first serialized pixel, WID Out(07:04). Refer to Table 2 on page 8 for the mapping of START values to nibbles on the serial data interface.

10.9 0x0008 Serializer Mode

7	6	5	4	3	2	1	0
RESERVED					MODE		

POR: 0x00

Bits 7 - 3: RESERVED

Bits 2- 0: MODE - determines the number of pixels to serialize before the serialization sequence resets.

000 2:1 Multiplex
001 4:1 Multiplex
010 8:1 Multiplex
011 16:1 Multiplex
100 Reserved
101 16:3 Multiplex
110 5:1 Multiplex
111 Reserved

10.10 0x0009 Pixel Interleave

7	6	5	4	3	2	1	0
RESERVED					INT		

POR: 0x00

Bits 7 - 3: RESERVED

Bits 2- 0: INT - Pixel Interleave. Equal to the number of pixels that the serialization sequence is rotated from line to line. Refer to Tables 6, 7 and 8 on page 12 for available interleave options.

Pixel interleave is only available for 4:1, 5:1, and 8:1 multiplex modes. If the serializer is set to any other mode, then the pixel interleave field, INT, must be set to zero for proper operation.

10.11 0x000A Miscellaneous Configuration

7	6	5	4	3	2	1	0
DDOT		XFER	PSIZE	LCI	WIDCTL		

POR: 0x00

Bits 7 - 6: DDOT - Determines the source of the divided dot clock output pin.

- 00 Pixel Clock
- 01 Pixel Clock / 2
- 10 Pixel Clock / 4
- 11 Pixel Clock / 8

Bit 5: XFER - Determines whether four or six transfers update the palette when the palette is being accessed as a ten bit palette.

- 0 Four data transfers are used to access one palette location.
- 1 Six data transfers are used to access one palette location.

This bit has no meaning when accessing the palette as eight bits (PSIZE = 0).

Bit 4: PSIZE - Determines whether the palette is accessed as eight bits per color band or ten bits per color band.

- 0 The palette is eight bits per color band. In this mode, the XFER bit has no affect. All palette updates are accomplished with three transfers.
- 1 The palette is ten bits per color band.

Bit 3: LCI - Load Clock Interleave Enable. Allows swapping of A and B buffer positions for 8 bit double buffered formats. The input pins swap positions after each load clock. The starting positions are the same for four lines, then swap for four lines.

- 0 Load Clock Interleave Disabled.
- 1 Load Clock Interleave Enabled.

See section 2.5, “Load Clock Interleave” on page 8 for detailed load clock interleave information.

Bits 2 - 0: WIDCTL - Window ID Out Control. Determines how data from the two serializer WID out fields, WID Out(07:04) and WID Out(03:00) is interpreted.

WIDCTL	Frame Buffer WID	Overlay WID	Overlay Data
000	0x0	0x0	Raw Pixel Data (31:16)
001	N/A	N/A	N/A
010	N/A	N/A	N/A
011	0x0	0x0	WID Out(07:00)
100	WID Out(03:00)	WID Out(03:00)	Raw Pixel Data(31:16)
101	WID Out(03:00)	WID Out(07:04)	Raw Pixel Data(31:16)
110	WID Out(03:00)	WID Out(03:00)	WID Out(07:04)
111	N/A	N/A	N/A

10.12 0x000B VGA Control

7	6	5	4	3	2	1	0
RESERVED					RDBK	PSIZE	VGA

POR: 0x00

Bits 7 - 3: RESERVED

Bit 2: RDBK - Readback Control. Determines what data is supplied during a read from processor I/O port 3.

0 The palette address (read mode) is returned by a read from port 3.

1 A status value representing the last type of palette access is returned. A value of 0x00 is returned if the last write to the palette address was for write mode (port 0). A value of 0x03 is returned if the last write to the palette address was for read mode (port 3).

Bit 1: PSIZE - Palette data port Size. Determines whether the palette data port (port 1) is accessed as 6 bits or 8 bits.

0 The palette is accessed as 6 bits per color band.

1 The palette is accessed as 8 bits per color band.

In either mode, the 6 or 8 bit value is linearized to 10 bits in the palette by copying some of the most significant bits of the input to the missing least significant bits in the palette.

Bit 0: VGA - Determines whether the RGB640 is operating in VGA mode.

0 VGA mode. The data from the VGA port of the serializer is used by the palette. The load clock input is used as the video clock for the internal pipeline.

1 VRAM mode. The PIX(159:000) inputs are latched with the load clock input and used for pixel and WID data. The video clock (dot clock) is used for clocking the internal pipeline.

10.13 0x000C DAC Compare / Monitor ID

7	6	5	4	3	2	1	0
MON				RSVD	RED	GRN	BLU

POR: 0x00

Bits 7 - 4: MON - Monitor ID input pins. This field is passed through from the four monitor ID input pins during a processor read operation of this register. The monitor ID pins are mapped such that the most significant, MON_ID(3), is presented on the most significant bit of the data bus, DATA(7), during the read access. In addition, MON_ID(2) is presented on DATA(6), MON_ID(1) on DATA(5), and MON_ID(0) on DATA(4).

Bit 3: RESERVED

Bit 2: RED - The result of a comparison of the red DAC output and the reference voltage (CVREF).

0 The red DAC output is greater than CVREF.

1 The red DAC output is less than CVREF.

Bit 1: GRN - The result of a comparison of the green DAC output and the reference voltage (CVREF).

0 The green DAC output is greater than CVREF.

1 The green DAC output is less than CVREF.

Bit 0: BLU - The result of a comparison of the blue DAC output and the reference voltage (CVREF).

0 The blue DAC output is greater than CVREF.

1 The blue DAC output is less than CVREF.

10.14 0x000D DAC Control

7	6	5	4	3	2	1	0
RESERVED				MONO	EN	SHUNT	SLEW

POR: 0x00

Bits 7 - 4: RESERVED

Bit 3: MONO - Monochrome Enable.

0 Red, green and blue DACs are all enabled. (The EN bit must also be set to ‘1’.)

1 Monochrome mode: the green DAC is enabled and the red and blue DACs are disabled.

Bit 2: EN - DAC Enable

0 All three DACs are disabled. The MONO bit has no effect.

1 All three DACs are enabled. (If the MONO bit is set to one, only the green DAC is enabled.)

Bit 1: SHUNT - Connects complementary DAC outputs to ground.

0 Normal operation.

1 Shunt the complementary DAC outputs to digital ground.

When the complimentary DAC outputs are unused, the best video signal quality is achieved by terminating the complimentary DAC outputs with an impedance that matches the total impedance on the positive DAC outputs and by setting the SHUNT bit to zero. An almost equal signal quality can be obtained by grounding the complimentary outputs or leaving them unconnected and by setting the SHUNT bit to one.

Bit 0: SLEW - DAC Output Slew Rate

0 Use fastest slew rate.

1 Limit (slow down) the DAC slew rate.

10.15 0x000E Update Control

7	6	5	4	3	2	1	0
RESERVED					RDAI	WRAI	WATCTL

POR: 0x00

Bits 7 - 3: RESERVED

Bit 2: RDAI - Read Access Auto Increment. Auto-increments the index register after a read access.

0 Enabled

1 Disabled

Bit 1: WRAI - Write Access Auto Increment. Auto-increments the index register after a write access.

0 Enabled

1 Disabled

Bit 0: WATCTL - WAT Update Control. Controls the timing of window attribute table updates.

0 Hold updates of the window attribute tables in temporary storage.

1 All updates of the window attribute table take effect during the next vertical retrace.

This function can be disabled by the Diagnostic Register (register 0x00FA). If a window attribute table is read, the values that are in effect, not the values in temporary storage, are returned.

10.16 0x000F Sync Control

7	6	5	4	3	2	1	0
RESERVED		PWR	VSP	HSP	CSE	CSG	BPE

POR: 0x00

Bits 7 - 6: RESERVED

Bit 5: PWR - Monitor Power Management.

0 Normal: sync pulses are passed through the RGB640 to the sync output pins and the DAC (for composite sync on green mode).

1 Power management: forces the vertical sync to the level of the VSP bit and forces the horizontal sync to the value of the HSP bit.

Bit 4: VSP - Vertical Sync Polarity. Determines whether the input vertical sync signal is to be inverted before being sent to the output or the DAC (for composite sync on green mode).

0 Normal (uninverted).

1 Inverted.

Bit 3: HSP - Horizontal Sync Polarity. Determines whether the input vertical sync signal is to be inverted before being sent to the output or the DAC (for composite sync on green mode).

0 Normal (uninverted).

1 Inverted.

Bit 2: CSE - Composite sync enable. Synthesizes a composite sync signal using the XOR of the vertical and horizontal sync signals. The vertical and horizontal sync inputs are processed using the PWR, VSP and HSP bits before the XOR is done.

0 Disables the formation of composite sync (that is, the horizontal sync input is used).

1 Enables the mixing of the vertical and horizontal sync signals. This composite sync waveform is output through the horizontal sync output pin and is used by the green DAC (for composite sync on green mode).

Bit 1: CSG - Composite Sync On Green. Controls whether the composite sync waveform is added to the green DAC output.

0 Adds no sync information to the green DAC output.

1 Adds the sync information to the green DAC output. The source of the sync information added to the green DAC output is determined by the CSE bit of this register.

Bit 0: BPE - Blanking Pedestal Enable. Controls whether the blanking level is different from the black level at the DAC outputs.

0 Disables the pedestal so that black and blank levels are the same.

1 Enables the pedestal so that black and blank levels differ from each other.

10.17 0x0010 Video PLL Reference Divide

7	6	5	4	3	2	1	0
RESERVED		N					

POR: 0x00

Bits 7 - 6: RESERVED

Bits 5 - 0: N - PLL reference divide. Sets the frequency division of the reference frequency of the video phase locked loop. The divide value is one greater than the N field in this register.

10.18 0x0011 Video PLL Multiplier

7	6	5	4	3	2	1	0
RSVD	M						

POR: 0x00

Bit 7: RESERVED

Bits 6 - 0: M - PLL frequency multiplier. Sets the multiplication value for the video PLL. The video PLL multiplies the divided reference frequency by one greater than the M field in this register.

$2 \leq M \leq 127$

$$video\ \ f_{VCO} = f_{VID\_REF} \times \frac{M+1}{N+1}$$

$video\ \ f_{VCO} \leq$ Maximum Video Clock Rate

$N =$ Register 0x0010

10.19 0x0012 Video PLL Output Divide

7	6	5	4	3	2	1	0
RESERVED				ODIV			

POR: 0x00

Bits 7 - 4: RESERVED

Bits 3 - 0: ODIV - PLL Output Divide. Selects the P divide value for the video PLL output.

ODIV	P Value	ODIV	P Value
0000	0	1000	4
0001	0	1001	4
0010	1	1010	Reserved
0011	1	1011	Reserved
0100	2	1100	Reserved
0101	2	1101	Reserved
0110	3	1110	Reserved
0011	3	1111	Reserved

$$video\ \ f_{CLOCKOUT} = f_{VID\_REF} \times \frac{M+1}{N+1}, P = 0$$

$$video\ \ f_{CLOCKOUT} = f_{VID\_REF} \times \frac{M+1}{(N+1) \times 2P}\ , P = 1, 2, 3, 4$$

N = Register 0x0010

M = Register 0x0011

10.20 0x0013 Video PLL Control

7	6	5	4	3	2	1	0
RESERVED				SEL	EN	C	

POR: 0x00

Bits 7 - 4: RESERVED

Bit 3: SEL - Pixel Clock Source Select.

0 The PLL is the source of the pixel clock.

1 The differential pixel clock input is the source of the pixel clock. ***When using this option, the ODIV field of the Video PLL Output Divide Register (register 0x0012) must be programmed with any value from 0010 to 1001 for proper operation***.

Bit 2: EN - Video PLL Enable.

0 Disabled

1 Enabled

Bits 1 - 0: C - Video PLL analog Control. Selects on-chip resistors to enhance the operation of the video PLL. This field should be set based on the video VCO operating frequency *video* f_{VCO}, not the divided down PLL frequency *video* $f_{CLOCKOUT}$.

00 Reserved

01 65 MHz $\leq$ *video* $f_{VCO} \leq$ 128 MHz

10 128 MHz $<$ *video* $f_{VCO} \leq$ 220 MHz

11 Reserved

10.21 0x0014 Auxiliary PLL Reference Divide

7	6	5	4	3	2	1	0
RESERVED		N					

POR: 0x00

Bits 7 - 6: RESERVED

Bits 5 - 0: N - PLL reference divide. Sets the frequency division of the reference frequency of the auxiliary phase locked loop. The divide value is one greater than the N field in this register.

To prevent the PLL from going out of range during the loading of PLL program data and assure a glitch free transition from the old to the new frequency of operation, a mechanism exists that updates the PLL internal programming values in parallel and in synchronism with the PLL output clock. The following register update sequence is required to engage this mechanism:

```
reg 0x0014 <- old N

reg 0x0014 <- new N

reg 0x0015 <- new M

reg 0x0016 <- new P

reg 0x0017 <- new C
```

The first write (writing 0x0014 without changing its value) engages the PLL update mechanism. The next three writes (new N, new M, new P) will be latched internally but will not be presented to the PLL. The final write (to reg 0x0017) will cause the PLL to synchronously update its programming values with the new N, M, P and C values.

The new values can not be read until they have taken effect following a write to the Auxiliary PLL Control Register (register 0x0017).

Refer to section 7.3.3, "Glitching on Frequency Change" on page 37 for more information on PLL frequency glitching.

10.22 0x0015 Auxiliary PLL Multiplier

7	6	5	4	3	2	1	0
RSVD	M						

POR: 0x03

Bit 7: RESERVED

Bits 6 - 0: M - PLL frequency multiplier. Sets the multiplication value for the auxiliary PLL. The auxiliary PLL multiplies the divided reference frequency by one greater than the M field in this register.

$2 \leq M \leq 127$

$$auxiliary\ \ f_{VCO} = f_{AUX\_REF} \times \frac{M+1}{N+1}$$

$$auxiliary\ \ f_{VCO} \leq \text{Maximum Auxiliary Clock Rate}$$

$$N = \text{Register 0x0014}$$

10.23 0x0016 Auxiliary PLL Output Divide

7	6	5	4	3	2	1	0
RESERVED				ODIV			

POR: 0x04

Bits 7 - 4: RESERVED

Bits 3 - 0: ODIV - PLL Output Divide. Selects the P divide value for the video PLL output.

ODIV	P Value	ODIV	P Value
0000	Reserved	1000	4
0001	Reserved	1001	4
0010	1	1010	Reserved
0011	1	1011	Reserved
0100	2	1100	Reserved
0101	2	1101	Reserved
0110	3	1110	Reserved
0011	3	1111	Reserved

$$f_{AUX\_PLL\_OUT} = f_{AUX\_REF} \times \frac{M+1}{(N+1) \times 2P}$$

N = Register 0x0014

M = Register 0x0015

10.24 0x0017 Auxiliary PLL Control

7	6	5	4	3	2	1	0
RESERVED					EN	C	

POR: 0x06

Bits 7 - 3: RESERVED

Bit 2: EN - Auxiliary PLL Enable.

0 Disabled

1 Enabled

Bits 1 - 0: C - Auxiliary PLL analog Control. Selects on-chip resistors to enhance the operation of the auxiliary PLL. This field should be set based on the auxiliary VCO operating frequency *auxiliary* f_{VCO}, not the divided down PLL frequency *auxiliary* $f_{AUXPLLOUT}$.

00 Reserved

01 65 MHz $\leq$ *auxiliary* $f_{VCO} \leq$ 128 MHz

10 128 MHz < *auxiliary* $f_{VCO} \leq$ 220 MHz

11 Reserved

10.25 0x0020 Chroma Key Register 0

7	6	5	4	3	2	1	0
KEY 0							

POR: 0x00

Bits 7 - 0: KEY0 - Chroma Key value 0. Chroma Key value 0 is compared to the overlay layer. If the overlay layer matches this value, the overlay layer becomes transparent (as long as chroma key mode is selected in the overlay window attribute table). The bits used in the comparison are determined by Chroma Key Mask 0 (register 0x0021).

10.26 0x0021 Chroma Key Mask 0

7	6	5	4	3	2	1	0
MASK 0							

POR: 0x00

Bits 7 - 0: MASK0 - Chroma Key Mask 0. Any bits of Chroma Key Mask 0 which are set to zero are not used to determine if the overlay layer matches Chroma Key value 0. Any bits of Chroma Key Mask 0 which are set to one force a comparison between the overlay layer and Chroma Key value 0 for that corresponding bit.

10.27 0x0022 Chroma Key Register 1

7	6	5	4	3	2	1	0
KEY 1							

POR: 0x00

Bits 7 - 0: KEY1 - Chroma Key value 1. Chroma Key value 1 is compared to the overlay layer. If the overlay layer matches this value, the overlay layer becomes transparent (as long as chroma key mode is selected in the overlay window attribute table). The bits used in the comparison are determined by Chroma Key Mask 1 (register 0x0023).

10.28 0x0023 Chroma Key Mask 1

7	6	5	4	3	2	1	0
MASK 1							

POR: 0x00

Bits 7 - 0: MASK1 - Chroma Key Mask 1. Any bits of Chroma Key Mask 1 which are set to zero are not used to determine if the overlay layer matches Chroma Key value 1. Any bits of Chroma Key Mask 1 which are set to one force a comparison between the overlay layer and Chroma Key value 1 for that corresponding bit.

10.29 0x0030 Cursor / Crosshair Control

7	6	5	4	3	2	1	0
RESERVED							PRI

POR: 0x00

Bits 7 - 1: RESERVED

Bit 0: PRI - Crosshair Priority. Controls whether the crosshair or the cursor is displayed when the two structures overlap.

0 The cursor is displayed when overlap occurs.

1 The crosshair is displayed when overlap occurs.

10.30 0x0031 Cursor Blink Rate

7	6	5	4	3	2	1	0
RATE							

POR: 0x00

Bits 7 - 0: RATE - Cursor Blink Rate. Controls the blink rate of both the crosshair and the cursor. A blink cycle lasts for a number of frames equal to the RATE value. Valid entries for RATE are 0x01 to 0xFF.

10.31 0x0032 Cursor Blink Duty Cycle

7	6	5	4	3	2	1	0
DUTY							

POR: 0x00

Bits 7 - 0: DUTY - Cursor Blink Duty Cycle. Controls the blink duty cycle of both the crosshair and the cursor. The primary color palette is used to display the crosshair and cursor from the beginning of the blink cycle for a number of frames equal to the DUTY value. After this time, either the alternate colors are displayed or the cursor and crosshair are transparent until the next blink cycle begins as determined by the Cursor Blink Rate Register (register 0x0031).

10.32 0x0040 Cursor Horizontal Position Lo

7	6	5	4	3	2	1	0
XLO							

POR: 0x00

Bits 7 - 0: XLO - contains the least significant 8 bits of the cursor horizontal position. The full cursor horizontal position is a 16 bit two's complement quantity describing the cursor position from the left side of the screen. The horizontal cursor position has a range of -4096 to +4095.

When not in diagnostic mode (as determined by the Diagnostics Register), the effect on the cursor by a write to this register is not seen immediately, and any read back of this register while the update is being held will return the previous value. See section 5.1.7 "Update" on page 31 and the description of the Diagnostics Register on page 82 for more details.

10.33 0x0041 Cursor Horizontal Position Hi

7	6	5	4	3	2	1	0
XHI							

POR: 0x00

Bits 7 - 0: XHI- contains the most significant 8 bits of the cursor horizontal position. The full cursor horizontal position is a 16 bit two's complement quantity describing the cursor position from the left side of the screen. The horizontal cursor position has a range of -4096 to +4095. ***Writing bits (6:4) has no effect. When read back, these bits return the value of the most significant bit, bit 7.***

When not in diagnostic mode (as determined by the Diagnostics Register), the effect on the cursor by a write to this register is not seen immediately, and any read back of this register while the update is being held will return the previous value. See section 5.1.7 "Update" on page 31 and the description of the Diagnostics Register on page 82 for more details.

10.34 0x0042 Cursor Vertical Position Lo

7	6	5	4	3	2	1	0
YLO							

POR: 0x00

Bits 7 - 0: YLO - contains the least significant 8 bits of the cursor vertical position. The full cursor vertical position is a 16 bit two's complement quantity describing the cursor position from the top of the screen. The vertical cursor position has a range of -4096 to +4095.

When not in diagnostic mode (as determined by the Diagnostics Register), the effect on the cursor by a write to this register is not seen immediately, and any read back of this register while the update is being held will return the previous value. See section 5.1.7 "Update" on page 31 and the description of the Diagnostics Register on page 82 for more details.

10.35 0x0043 Cursor Vertical Position Hi

7	6	5	4	3	2	1	0
YHI							

POR: 0x00

Bits 7 - 0: YHI - contains the least significant 8 bits of the cursor vertical position. The full cursor vertical position is a 16 bit two's complement quantity describing the cursor position from the top of the screen. The vertical cursor position has a range of -4096 to +4095. ***Writing bits (6:4) has no effect. When read back, these bits return the value of the most significant bit, bit 7.***

When not in diagnostic mode (as determined by the Diagnostics Register), the effect on the cursor by a write to this register is not seen immediately, and any read back of this register while the update is being held will return the previous value. See section 5.1.7 "Update" on page 31 and the description of the Diagnostics Register on page 82 for more details.

10.36 0x0044 Cursor Horizontal Offset

7	6	5	4	3	2	1	0
RESERVED		XOFF					

POR: 0x00

Bits 7 - 6: RESERVED

Bits 5 - 0: XOFF - Cursor Horizontal Offset. Represents the horizontal location of the pixel within the cursor pixel map that is used to position the cursor on the screen. An XOFF value of 0x00 places the left side of the cursor at the position specified by the Cursor Position Registers. An XOFF value of 0x3F places the right side of the cursor at the position specified by the Cursor Position Registers when the cursor is in 64 x 64 pixel mode.

When not in diagnostic mode (as determined by the Diagnostics Register), the effect on the cursor by a write to this register is not seen immediately, and any read back of this register while the update is being held will return the previous value. See section 5.1.7 "Update" on page 31 and the description of the Diagnostics Register on page 82 for more details.

10.37 0x0045 Cursor Vertical Offset

7	6	5	4	3	2	1	0
RESERVED		YOFF					

POR: 0x00

Bits 7 - 6: RESERVED

Bits 5 - 0: YOFF - Cursor Vertical Offset. Represents the vertical location of the pixel within the cursor pixel map that is used to position the cursor on the screen. A YOFF value of 0x00 places the top of the cursor at the position specified by the Cursor Position Registers. A YOFF value of 0x3F places the bottom of the cursor at the position specified by the Cursor Position Registers when the cursor is in 64 x 64 pixel mode.

When not in diagnostic mode (as determined by the Diagnostics Register), the effect on the cursor by a write to this register is not seen immediately, and any read back of this register while the update is being held will return the previous value. See section 5.1.7 "Update" on page 31 and the description of the Diagnostics Register on page 82 for more details.

10.38 0x0046 Advanced Function Cursor Color 0

7	6	5	4	3	2	1	0
RESERVED					RED	GRN	BLU

POR: 0x00

Bits 7 - 3: RESERVED

Bit 2: RED - For the advanced cursor mode, this bit is substituted as the most significant bit for the red color band when the translucent cursor is being used and cursor color 0 is being selected.

Bit 1: GRN - For the advanced cursor mode, this bit is substituted as the most significant bit for the green color band when the translucent cursor is being used and cursor color 0 is being selected.

Bit 0: BLU - For the advanced cursor mode, this bit is substituted as the most significant bit for the blue color band when the translucent cursor is being used and cursor color 0 is being selected.

When not in diagnostic mode (as determined by the Diagnostics Register), the effect on the cursor by a write to this register is not seen immediately, and any read back of this register while the update is being held will return the previous value. See section 5.1.7 "Update" on page 31 and the description of the Diagnostics Register on page 82 for more details.

10.39 0x0047 Advanced Function Cursor Color 1

7	6	5	4	3	2	1	0
RESERVED					RED	GRN	BLU

POR: 0x00

Bits 7 - 3: RESERVED

Bit 2: RED - For the advanced cursor mode, this bit is substituted as the most significant bit for the red color band when the translucent cursor is being used and cursor color 1 is being selected.

Bit 1: GRN - For the advanced cursor mode, this bit is substituted as the most significant bit for the green color band when the translucent cursor is being used and cursor color 1 is being selected.

Bit 0: BLU - For the advanced cursor mode, this bit is substituted as the most significant bit for the blue color band when the translucent cursor is being used and cursor color 1 is being selected.

When not in diagnostic mode (as determined by the Diagnostics Register), the effect on the cursor by a write to this register is not seen immediately, and any read back of this register while the update is being held will return the previous value. See section 5.1.7 "Update" on page 31 and the description of the Diagnostics Register on page 82 for more details.

10.40 0x0048 Advanced Function Cursor Color 2

7	6	5	4	3	2	1	0
RESERVED					RED	GRN	BLU

POR: 0x00

Bits 7 - 3: RESERVED

Bit 2: RED - For the advanced cursor mode, this bit is substituted as the most significant bit for the red color band when the translucent cursor is being used and cursor color 2 is being selected.

Bit 1: GRN - For the advanced cursor mode, this bit is substituted as the most significant bit for the green color band when the translucent cursor is being used and cursor color 2 is being selected.

Bit 0: BLU - For the advanced cursor mode, this bit is substituted as the most significant bit for the blue color band when the translucent cursor is being used and cursor color 2 is being selected.

When not in diagnostic mode (as determined by the Diagnostics Register), the effect on the cursor by a write to this register is not seen immediately, and any read back of this register while the update is being held will return the previous value. See section 5.1.7 "Update" on page 31 and the description of the Diagnostics Register on page 82 for more details.

10.41 0x0049 Advanced Function Cursor Color 3

7	6	5	4	3	2	1	0
RESERVED					RED	GRN	BLU

POR: 0x00

Bits 7 - 3: RESERVED

Bit 2: RED - For the advanced cursor mode, this bit is substituted as the most significant bit for the red color band when the translucent cursor is being used and cursor color 3 is being selected.

Bit 1: GRN - For the advanced cursor mode, this bit is substituted as the most significant bit for the green color band when the translucent cursor is being used and cursor color 3 is being selected.

Bit 0: BLU - For the advanced cursor mode, this bit is substituted as the most significant bit for the blue color band when the translucent cursor is being used and cursor color 3 is being selected.

When not in diagnostic mode (as determined by the Diagnostics Register), the effect on the cursor by a write to this register is not seen immediately, and any read back of this register while the update is being held will return the previous value. See section 5.1.7 "Update" on page 31 and the description of the Diagnostics Register on page 82 for more details.

10.42 0x004A Advanced Function Cursor Attribute Table

7	6	5	4	3	2	1	0
ATT3		ATT2		ATT1		ATT0	

POR: 0x00

Bits 7 - 6: ATT3 - Attribute 3 bits for the advanced function cursor. This field determines the display attribute (transparent, solid, translucent, or highlighted) of any pixel in the cursor pixel map whose value is b'11'. This field has no effect if the advanced function of the cursor is not being used. Table 21 on page 30 details how this two bit field is decoded.

Bits 5 - 4: ATT2 - Attribute 2 bits for the advanced function cursor. This field determines the display attribute (transparent, solid, translucent, or highlighted) of any pixel in the cursor pixel map whose value is b'10'. This field has no effect if the advanced function of the cursor is not being used. Table 21 on page 30 details how this two bit field is decoded.

Bits 3 - 2: ATT1 - Attribute 1 bits for the advanced function cursor. This field determines the display attribute (transparent, solid, translucent, or highlighted) of any pixel in the cursor pixel map whose value is b'01'. This field has no effect if the advanced function of the cursor is not being used. Table 21 on page 30 details how this two bit field is decoded.

Bits 1 - 0: ATT0 - Attribute 0 bits for the advanced function cursor. This field determines the display attribute (transparent, solid, translucent, or highlighted) of any pixel in the cursor pixel map whose value is b'00'. This field has no effect if the advanced function of the cursor is not being used. Table 21 on page 30 details how this two bit field is decoded.

When not in diagnostic mode (as determined by the Diagnostics Register), the effect on the cursor by a write to this register is not seen immediately, and any read back of this register while the update is being held will return the previous value. See section 5.1.7 "Update" on page 31 and the description of the Diagnostics Register on page 82 for more details.

10.43 0x004B Cursor Control

7	6	5	4	3	2	1	0
PAR		BLINK	BTT	SIZE	MODE		

POR: 0x00

Bits 7 - 6: PAR - Partition. Selects one of four partitions within the cursor pixel map for display. This field is used only when the cursor is 32 x 32 pixels large.

PAR	Update Index	Read Back Index
00	0x1000 - 0x10FF	0x2000 - 0x20FF
01	0x1100 - 0x11FF	0x2100 - 0x21FF
10	0x1200 - 0x12FF	0x2200 - 0x22FF
11	0x1200 - 0x12FF	0x2300 - 0x23FF

Bit 5: BLINK - Cursor Blinking Control.

0 Cursor does not blink.

1 Cursor blinking is enabled.

Bit 4: BTT - Blink To Transparent. Determines whether the cursor blinks to transparent or to the set of alternate cursor colors during the second part of the blink cycle.

0 Alternate cursor colors are used.

1 Cursor blinks to transparent.

Cursor blinking must be enabled for the BTT bit to have any effect.

Bit 3: SIZE - Determines the size of the cursor.

0 32 x 32

1 64 x 64

Bits 2 - 0: MODE The cursor mode field determines how the cursor will operate. See Table 20 on page 30 for a complete description of these modes.

MODE	Cursor Mode
000	Off
001	Mode 0
010	Mode 1
011	Mode 2
100	Advanced
101	Reserved
110	Reserved
111	Reserved

When not in diagnostic mode (as determined by the Diagnostics Register), the effect on the cursor by a write to this register is not seen immediately, and any read back of this register while the update is being held will return the previous value. See section 5.1.7 "Update" on page 31 and the description of the Diagnostics Register on page 82 for more details.

10.44 0x0050 Crosshair Horizontal Position Lo

7	6	5	4	3	2	1	0
XLO							

POR: 0x00

Bits 7 - 0: XLO - contains the least significant 8 bits of the crosshair horizontal position. The full crosshair horizontal position is a 16 bit two's complement quantity describing the crosshair position from the left side of the screen. The horizontal crosshair position has a range of -4096 to +4095.

When not in diagnostic mode (as determined by the Diagnostics Register), the effect on the crosshair position by a write to this register is not seen immediately, and any read back of this register while the update is being held will return the previous value. See section 5.2.2 "Position" on page 32 and the description of the Diagnostics Register on page 82 for more details.

10.45 0x0051 Crosshair Horizontal Position Hi

7	6	5	4	3	2	1	0
XHI							

POR: 0x00

Bits 7 - 0: XHI - contains the most significant 8 bits of the crosshair horizontal position. The full crosshair horizontal position is a 16 bit two's complement quantity describing the crosshair position from the left side of the screen. The horizontal crosshair position has a range of -4096 to +4095. ***Writing bits (6:4) has no effect. When read back, these bits return the value of the most significant bit, bit 7.***

When not in diagnostic mode (as determined by the Diagnostics Register), the effect on the crosshair position by a write to this register is not seen immediately, and any read back of this register while the update is being held will return the previous value. See section 5.2.2 "Position" on page 32 and the description of the Diagnostics Register on page 82 for more details.

10.46 0x0052 Crosshair Vertical Position Lo

7	6	5	4	3	2	1	0
YLO							

POR: 0x00

Bits 7 - 0: YLO - contains the least significant 8 bits of the crosshair vertical position. The full crosshair vertical position is a 16 bit two's complement quantity describing the crosshair position from the top of the screen. The vertical crosshair position has a range of -4096 to +4095.

When not in diagnostic mode (as determined by the Diagnostics Register), the effect on the crosshair position by a write to this register is not seen immediately, and any read back of this register while the update is being held will return the previous value. See section 5.2.2 "Position" on page 32 and the description of the Diagnostics Register on page 82 for more details.

10.47 0x0053 Crosshair Vertical Position Hi

7	6	5	4	3	2	1	0
YHI							

POR: 0x00

Bits 7 - 0: YHI - contains the least significant 8 bits of the crosshair vertical position. The full crosshair vertical position is a 16 bit two's complement quantity describing the crosshair position from the top of the screen. The vertical crosshair position has a range of -4096 to +4095. ***Writing bits (6:4) has no effect. When read back, these bits return the value of the most significant bit, bit 7.***

When not in diagnostic mode (as determined by the Diagnostics Register), the effect on the crosshair position by a write to this register is not seen immediately, and any read back of this register while the update is being held will return the previous value. See section 5.2.2 "Position" on page 32 and the description of the Diagnostics Register on page 82 for more details.

10.48 0x0054 Crosshair Pattern Color

7	6	5	4	3	2	1	0
RESERVED		FILL		BORD		OUT	

POR: 0x00

Bits 7 - 6: RESERVED

Bits 5 - 4: FILL - Defines the fill color for the patterned crosshair. The fill area is at the center of the crosshair. This field has no effect when the monochrome crosshair is selected.

FILL	Fill Color
00	Transparent
01	Crosshair Color 1
10	Crosshair Color 2
11	Crosshair Color 3

Bits 3 - 2: BORD - Defines the border color for the patterned crosshair. The border area is the outermost area of the patterned crosshair. This field has no effect when the monochrome crosshair is selected.

BORD	Border Color
00	Transparent
01	Crosshair Color 1
10	Crosshair Color 2
11	Crosshair Color 3

Bits 1 - 0: OUT - Defines the outline color for the patterned crosshair. The outline area is the area between the border and fill areas of the patterned crosshair. This field has no effect when the monochrome crosshair is selected.

OUT	Outline Color
00	Transparent
01	Crosshair Color 1
10	Crosshair Color 2
11	Crosshair Color 3

10.49 0x0055 Crosshair Vertical Pattern

7	6	5	4	3	2	1	0
RSVD	YPAT						

POR: 0x00

Bit 7: RESERVED

Bits 6 - 0: YPAT - Y Pattern. Defines the colors of the vertical cross section of the horizontal leg of the patterned crosshair. For a crosshair width of 7 pixels, bit 6 of the YPAT field determines the color to choose for the top pixel of the horizontal leg and bit 0 determines the color to choose for the bottom pixel. Any bits set to one in the YPAT field force the corresponding pixel in the horizontal leg of the crosshair to be the outline color (as selected by the Crosshair Pattern Color Register, register 0x0054). Any bits of the YPAT field that are set to zero and are enclosed by the outline color force the corresponding pixels to be the fill color. Any other bits of the YPAT field that are set to zero force the corresponding pixels to be the border color. Patterns with more than one contiguous fill area cannot be used (for example, b'0101010'). For widths less than 7 pixels, the full 7 bit pattern is used and the result is then clipped to the requested width by removing the outermost pixels.

Refer to the description of the Crosshair Horizontal Pattern Register for examples of values and the resulting patterns.

10.50 0x0056 Crosshair Horizontal Pattern

7	6	5	4	3	2	1	0
RSVD	XPAT						

POR: 0x00

Bit 7: RESERVED

Bits 6 - 0: XPAT - X Pattern. Defines the colors of the horizontal cross section of the vertical leg of the patterned crosshair. For a crosshair width of 7 pixels, bit 6 of the XPAT field determines the color to choose for the leftmost pixel of the vertical leg and bit 0 determines the color to choose for the rightmost pixel. Any bits set to one in the XPAT field force the corresponding pixel in the vertical leg of the crosshair to be the outline color (as selected by the Crosshair Pattern Color Register, register 0x0054). Any bits of the XPAT field that are set to zero and are enclosed by the outline color force the corresponding pixels to be the fill color. Any other bits of the XPAT field that are set to zero force the corresponding pixels to be the border color. Patterns with more than one contiguous fill area cannot be used (for example, b'0101010').

XPAT	Pattern
0000000	BBBBBBB
0011100	BBOOOBB
0110110	BOOFOOB
1100011	OOFFFOO
1111111	OOOOOOO

For widths less than 7 pixels, the full 7 bit pattern is used and the result is then clipped to the requested width by removing the outermost pixels. As an example, for a 3 bit wide crosshair, the pattern of b'1100011' would force the 3 pixels of the vertical leg to the fill color.

10.51 0x0057 Crosshair Control 1

7	6	5	4	3	2	1	0
FILL	WIDTH		CLIP	BLINK	COLOR		PEN

POR: 0x00

Bit 7: FILL - Determines the display priority when outline and fill pixels of the patterned crosshair overlap.

0 Outline color is displayed wherever it overlays the fill color.

1 Fill color is displayed wherever is overlays the outline color.

The border color has the lowest display priority of the three patterned colors and is not shown wherever it is overlaid by the outline or fill colors.

Bits 6 - 5: WIDTH - Sets the crosshair width.

WIDTH	Crosshair Width
00	1 pixel
01	3 pixels
10	5 pixels
11	7 pixels

Bit 4: CLIP - Determines whether the crosshair is full screen or can be clipped to a logical window.

0 Window clipping is disabled and a full screen crosshair is displayed when enabled and on screen.

1 Window clipping is enabled and the crosshair is shown only in those windows whose ID allows crosshair display, as determined by the frame buffer and overlay window attribute tables.

Bit 3: BLINK - Controls the blinking of the crosshair.

0 Crosshair does not blink.

1 Crosshair blinking is enabled.

Bits 2 - 1: COLOR- Selects the crosshair color to use when the monochrome crosshair is selected. This field has no effect when using the patterned crosshair.

FILL	Fill Color
00	Transparent
01	Crosshair Color 1
10	Crosshair Color 2
11	Crosshair Color 3

Bit 0: PEN - Determines whether the crosshair is monochrome or patterned.

0 Crosshair is monochrome.

1 The three color patterned crosshair is enabled.

To disable the crosshair function completely, write 0x00 to this register.

10.52 0x0058 Crosshair Control 2

7	6	5	4	3	2	1	0
RESERVED							BTT

POR: 0x00

Bits 7 - 1: RESERVED

Bit 0: BTT - Blink To Transparent. Determines whether the crosshair blinks to transparent or to the set of alternate crosshair colors during the second part of the blink cycle.

0 Alternate crosshair colors are used.

1 Crosshair blinks to transparent.

Crosshair blinking must be enabled for the BTT bit to have any effect.

10.53 0x0070 YUV Conversion Coefficient K1

7	6	5	4	3	2	1	0
K1							

POR: 0x00

Bits 7 - 0: K1 - YUV to RGB Converter Constant.

10.54 0x0071 YUV Conversion Coefficient K2

7	6	5	4	3	2	1	0
K2							

POR: 0x00

Bits 7 - 0: K2 - YUV to RGB Converter Constant.

10.55 0x0072 YUV Conversion Coefficient K3

7	6	5	4	3	2	1	0
K3							

POR: 0x00

Bits 7 - 0: K3 - YUV to RGB Converter Constant.

10.56 0x0073 YUV Conversion Coefficient K4

7	6	5	4	3	2	1	0
K4							

POR: 0x00

Bits 7 - 0: K4 - YUV to RGB Converter Constant.

The register values K1, K2, K3 and K4 provide the constants for the programmable YUV to RGB converter, as discussed in Sections 3.3.1 “Color Space Conversion” on page 15 and 3.3.4 “Coefficients” on page 16.

10.57 0x00F0 VRAM Mask Register 0

7	6	5	4	3	2	1	0
MASK07	MASK06	MASK05	MASK04	MASK03	MASK02	MASK01	MASK00

POR: 0x00

Bit 7: MASK07 - Masks serial data inputs PIX(063:056).

Bit 6: MASK06 - Masks serial data inputs PIX(055:048).

Bit 5: MASK05 - Masks serial data inputs PIX(047:040).

Bit 4: MASK04 - Masks serial data inputs PIX(039:032).

Bit 3: MASK03 - Masks serial data inputs PIX(031:024).

Bit 2: MASK02 - Masks serial data inputs PIX(023:016).

Bit 1: MASK01 - Masks serial data inputs PIX(015:008).

Bit 0: MASK00 - Masks serial data inputs PIX(007:000).

A value of 0 for any of the bits in this register masks (forces to 0) the corresponding received serial data inputs. Setting the bit to 1 enables the serial data inputs to be received and used normally as pixel and WID data.

10.58 0x00F1 VRAM Mask Register 1

7	6	5	4	3	2	1	0
MASK15	MASK14	MASK13	MASK12	MASK11	MASK10	MASK09	MASK08

POR: 0x00

Bit 7: MASK15 - Masks serial data inputs PIX(127:120).

Bit 6: MASK14 - Masks serial data inputs PIX(119:112).

Bit 5: MASK13 - Masks serial data inputs PIX(111:104).

Bit 4: MASK12 - Masks serial data inputs PIX(103:096).

Bit 3: MASK11 - Masks serial data inputs PIX(095:088).

Bit 2: MASK10 - Masks serial data inputs PIX(087:080).

Bit 1: MASK09 - Masks serial data inputs PIX(079:072).

Bit 0: MASK08 - Masks serial data inputs PIX(071:064).

A value of 0 for any of the bits in this register masks (forces to 0) the corresponding received serial data inputs. Setting the bit to 1 enables the serial data inputs to be received and used normally as pixel and WID data.

10.59 0x00F2 VRAM Mask Register 2

7	6	5	4	3	2	1	0
RESERVED				MASK19	MASK18	MASK17	MASK16

POR: 0x00

Bits 7 - 4: RESERVED

Bit 3: MASK19 - Masks serial data inputs PIX(159:152).

Bit 2: MASK18 - Masks serial data inputs PIX(151:144).

Bit 1: MASK17 - Masks serial data inputs PIX(143:136).

Bit 0: MASK16 - Masks serial data inputs PIX(135:128).

A value of 0 for any of the bits in this register masks (forces to 0) the corresponding received serial data inputs. Setting the bit to 1 enables the serial data inputs to be received and used normally as pixel and WID data.

10.60 0x00FA Diagnostics

7	6	5	4	3	2	1	0
RESERVED					XHRIM	CURIM	WATIM

POR: 0x00

Bits 7 - 3: RESERVED

Bit 2: XHRIM - Crosshair Immediate Update. Allows the Crosshair Position Registers (registers 0x0050 to 0x0053) to be updated immediately.

0 Crosshair position updates are held off until the next vertical retrace. However, when Crosshair Horizontal Position Lo (register 0x0050) is written, all crosshair position updates are held off completely. Updates can be resumed again by witing to Crosshair Vertical Position Hi (register 0x0053). The first update will occur at the following vertical retrace.

1 Crosshair position is updated immediately as each position register is written.

Bit 1: CURIM - Cursor Immediate Update. Allows all Cursor Registers (registers 0x0040 to 0x004B) to be updated immediately.

0 Cursor register updates are held off until the next vertical retrace. However, when Cursor Horizontal Position Lo (register 0x0040) is written, all cursor register updates are held off completely. Updates can be resumed again by writng to Cursor Vertical Position Hi (register 0x0043). The first update will occur at the following vertical retrace.

1 Cursor registers are updated immediately as each register is written.

Bit 0: WATIM - Window Attribute Immediate Update. Allows the overlay and frame buffer window attribute tables to be updated immediately.

0 Window attribute table updates are held off under the guidance of the Update Control Register (register 0x000E).

1 Window attribute updates occur immediately as each is written to the processor port.

10.61 0x00FB MISR Control / Status

7	6	5	4	3	2	1	0
RESERVED					STATUS		EN

POR: 0x00

Bits 7 - 3: RESERVED

Bits 2 - 1: STATUS - Contains the state of the MISR signature collection.

00 Reset

01 Signature Collection Complete

10 Signature Collection in Progress

11 Invalid

Bit 0: EN - MISR Enable. Setting this bit to one starts signature collection at the next active frame. To capture another signature, the MISR Enable bit must be explicitly set to zero for one full frame. Another signature collection may be started on the following frame.

10.62 0x00FC MISR Signature 0

7	6	5	4	3	2	1	0
SIG0							

POR: 0x00

Bits 7 - 0: SIG0 - MISR signature bits 07:00.

10.63 0x00FD MISR Signature 1

7	6	5	4	3	2	1	0
SIG1							

POR: 0x00

Bits 7 - 0: SIG1 - MISR signature bits 15:08.

10.64 0x00FE MISR Signature 2

7	6	5	4	3	2	1	0
SIG2							

POR: 0x00

Bits 7 - 0: SIG2 - MISR signature bits 23:16.

10.65 0x00FF MISR Signature 3

7	6	5	4	3	2	1	0
RESERVED		SIG3					

POR: 0x3F

Bits 7- 6: RESERVED

Bits 5- 0: SIG3 - MISR signature bits 29:24.

The MISR signature is a 30 bit result of the MISR operating for one complete frame. The contents of these registers are the inverse of the actual signature. See Section 8.1 on page 39 for more details.

11.0 Pin Descriptions

11.1 Summary

Table 33. Pin Summary

Description	Type	Number
Video Reference Clock	Input	1
External Video Clock	Input	2
Auxiliary Reference Clock	Input	1
Auxiliary PLL Output Clock	Output	1
Divided Dot Clock	Output	1
Serial Clock	Output	1
Load Clock	Input	1
Pixel Data	Input	160
Blank	Input	1
Horizontal Sync Input	Input	1
Vertical Sync Input	Input	1
Horizontal Sync Output	Output	1
Vertical Sync Output	Output	1
Processor Port Control	Input	5
Processor Data Bus	I/O	8
Reset	Input	1
Monitor Identification	Input	4
Test	Input	9
DAC Support	Analog	10
PLL Support	Analog	4
Total		214

11.2 Detail

Table 34. Pin Description

Signal	Type	QFP Pin(s)	BGA Pin(s)	Description
Clock and Clock Control Pins				
VID_REF	I	212	M02	Video Reference Clock. A fixed frequency of 2 MHz to 100 MHz applied to this pin provides the reference clock for the programmable video PLL. If this input is not used, it must be terminated to digital V_{DD}, preferably through a 10KΩ resistor.
VID_EXT_CLK $\overline{\text{VID\_EXT\_CLK}}$	I I	245 246	U03 T03	External Video Clock. This pair of differential ECL receivers can be used to supply the video clock when the programmable video PLL is not used. Otherwise, if these inputs are not used, they must be terminated, VID_EXT_CLK to digital V_{DD} preferably with a 10KΩ resistor, and $\overline{\text{VID\_EXT\_CLK}}$ to digital GND preferably with a 75Ω resistor.
AUX_REF	I	252	T05	Auxiliary Reference Clock. A fixed frequency of 2 MHz to 100 MHz applied to this pin provides the reference clock for the programmable auxiliary PLL. If this input is not used, it must be terminated to digital V_{DD}, preferably with a 10KΩ resistor.
AUX_PLL_OUT	O	099	F15	Auxiliary PLL Output Clock. The output of the programmable auxiliary PLL is provided by this pin.
DDOTCLK	O	032	V15	Divided Dot Clock. This pin supplies the internal video clock divided by 1, 2, 4, or 8. The source of the internal video clock can be the output of the programmable video PLL, the external video clock, or the load clock (when the chip is in VGA mode). The divide factor is under the control of the Miscellaneous Configuration Register (register 0x000A). This output can supply a clock speed not to exceed 100 MHz.
SCLK	O	174	B01	Serial Clock. This pin supplies the divided version of the internal video clock needed to clock pixel data into the serial data port. The divide factor is determined by the setting of the Serializer Mode Register (register 0x0008). For example, when in 4:1 multiplex mode, the internal video clock is divided by 4 to form the serial clock. The serial clock output should not be used when the chip is in VGA mode as determined by the VGA Control Register (register 0x000B).
LCLK	I	077	K13	Load Clock. The rising edge of load clock latches pixel data and monitor timing signals (blank, horizontal sync, and vertical sync) into the chip.

Table 34. Pin Description (Continued)

Signal	Type	QFP Pin(s)	BGA Pin(s)	Description
Video Data Input Pins				
PIX(159:128)	I	128, 215, 013, 081, 148, 213, 011, 079, 145, 207, 005, 073, 139, 210, 265, 076, 142, 198, 268, 063, 130, 199, 269, 065, 131, 194, 264, 059, 126, 191, 261, 056,	G09, L05, N10, H12, F06, K06, M09, J11, H08, M07, T09, K11, G08, N07, P08, J10, D09, P06, T08, K10, E10, L07, V09, L10, F10, K07, M08, M10, H09, J06, R07, N12	Pixel Data. This bus is the interface between the frame buffer and the chip. These inputs are latched by the rising edge of load clock. Unused inputs on the pixel data bus may be left unconnected. All inputs on this bus are internally pulled up with an active load which has an effective resistance of 15KΩ.
PIX(127:64)	I	094, 172, 253, 042, 124, 186, 258, 053, 092, 170, 257, 046, 132, 188, 259, 054, 097, 171, 256, 044, 129, 189, 260, 055, 096, 173, 018, 061, 133, 193, 263, 057, 153, 175, 017, 047, 138, 192, 262, 062, 157, 176, 021, 051, 137, 196, 266, 064, 159, 180, 022, 050, 141, 197, 267, 067, 161, 178, 023, 052, 140, 200, 270, 066,	F13, D03, U06, U16, D11, G02, P07, N16, D10, C02, T06, W16, B08, H05, T07, M11, F16, F03, V06, P15, C09, J04, W06, L11, E11, E03, V11, L12, F09, H01, V07, M15, B05, G05, T11, R16, F08, J03, U07, M14, C05, C01, W12,M12, C08, K03, N08, L16, A04, H06, T12, P16, A07, J02, W07, K16, E05, G04, V12, N14, B07, J01, W08, L14	Pixel Data continued.

Table 34. Pin Description (Continued)

Signal	Type	QFP Pin(s)	BGA Pin(s)	Description
PIX(63:00)	I	160, 177, 008, 084, 146, 201, 271, 072, 158, 179, 009, 086, 144, 206, 004, 071, 162, 181, 026, 085, 147, 205, 003, 075, 163, 182, 024, 089, 149, 209, 007, 074, 165, 183, 028, 090, 150, 208, 006, 080, 167, 187, 029, 091, 151, 214, 012, 078, 168, 211, 041, 093, 152, 217, 015, 082, 166, 216, 033, 095, 154, 218, 016, 083	D05, E02, U09, J07, D07, K05, V08, J16, H07, D01, P10, H14, C07, K01, W09, J14, C04, E01, V14, G13, A06, L01, R09, H16, B03, F01, P11, G16, E07, L02, W10, K15, D04, G03, W14, F14, G06, L03, N09, K14, F05, H03, T13, G15, C06, M01, T10, J12, B02, L04, R15, G14, F07, M03, U11, H15, A02, N01, U15, G12, A05, N03, R11, H13	Pixel Data continued.

Table 34. Pin Description (Continued)

Signal	Type	QFP Pin(s)	BGA Pin(s)	Description
Video Control Pins				
$\overline{\text{BLANK}}$	I	100	H11	Composite Blank. This input should be held low during horizontal and vertical blanking periods. It should be held high during pixel display periods. This input is latched by the rising edge of load clock.
HSYNC_IN	I	164	A03	Horizontal Sync Input. This input can be used as the horizontal sync or the composite sync input. When used as horizontal sync, it can be combined with vertical sync internally to form composite sync by programming the Sync Control Register (register 0x000F). In both modes, this sync input is latched by the rising edge of load clock and pipelined through the chip to the DAC or the horizontal sync output. If this input is not used, it must be terminated to digital V_{DD}, preferably with a 10KΩ resistor.
VSYNC_IN	I	190	G01	Vertical Sync Input. This input can be combined with horizontal sync internally to form composite sync by programming the Sync Control Register (register 0x000F). The vertical sync input is latched by the rising edge of load clock and pipelined through the chip to the DAC or the vertical sync output. If this input is not used, it must be terminated to digital V_{DD}, preferably with a 10KΩ resistor.
HSYNC_OUT	O	010	V10	Horizontal Sync Output. This output can be configured as a monitor horizontal sync or a monitor composite sync by the Sync Control Register (register 0x000F). The Sync Control Register determines the polarity of the horizontal sync output. This register can also force the output to a specified level.
VSYNC_OUT	O	014	W11	Vertical Sync Output. The Sync Control Register (register 0x000F) determines the polarity of the vertical sync output. This register can also force the output to a specified level.
Processor Interface Pins				
$\overline{\text{CE\_RD}}$	I	058	M13	Read Strobe. When read strobe is held low, the register contents selected by RS(2:0) are driven onto the processor data bus, DATA(7:0). The register select pins, RS(2:0), must be valid at the falling edge of the read strobe. Data is driven onto the processor bus as long as read strobe is held low. The read strobe should not be allowed to float when connected to a 3-state net. The net should be pulled up to avoid unpredictable results.
$\overline{\text{CE\_WT}}$	I	060	M16	Write Strobe. The write strobe allows data on the processor data bus, DATA(7:0), to be written to the register selected by RS(2:0). The register select pins, RS(2:0), must be valid at the falling edge of the write strobe. The values on the processor data bus, DATA(7:0), must be valid at the rising edge of the write strobe. The write strobe should not be allowed to float when connected to a 3-state net. The net should be pulled up to avoid unpredictable results.
RS(2:0)	I	195, 127, 143	J05, E09, G07	Register Selects. These bits are used to select one of the eight registers in the processor I/O space. These values must be valid at the falling edge of $\overline{\text{CE\_RD}}$ during read operations and at the falling edge of $\overline{\text{CE\_WT}}$ during write operations. Refer to section section 9.1, “Processor I/O Space” on page 40 for an address map of these locations and a description of their function.
DATA(7:0)	I/O	045, 043, 040, 039, 027, 025, 031, 030	T16, P14, V16, P13, U13, W13, R13, W15	Processor Data Bus. These bits are the bidirectional data bus used to read and write internal registers. The drivers are enabled when $\overline{\text{CE\_RD}}$ is held low. At all other times, the drivers are 3-stated.

Table 34. Pin Description (Continued)

Signal	Type	QFP Pin(s)	BGA Pin(s)	Description
Miscellaneous Pins				
N_RESET	I	250	R05	Reset. Reset, in its active low state, sets all registers to the power on state listed in section section 10.0, “Register Descriptions” on page 48. A minimum 1 µs pulse width is required to reset registers. The color palettes and cursor pixel map are not initialized to a predetermined value. Reset is required after power on to guarantee proper PLL operation.
MON_ID(3:0)	I	251, 220, 248, 249	U05, M04, U04, T04	Monitor Identification. This four bit value can be read from the processor bus. It is intended to aid in the determination of the type of monitor in use. The DAC Compare / Monitor ID Register (register 0x000C) can be accessed to read this value.
DAC Output Pins				
RED GREEN BLUE	O O O	107 110 121	A12 B13 A15	Red, green, and blue analog outputs. These analog monitor outputs are RS-343A compatible, have internal clamping and flash-over protection, and are capable of driving doubly terminated 75Ω or 100Ω coax without buffering or external components.
$\overline{\text{RED}}$ $\overline{\text{GREEN}}$ $\overline{\text{BLUE}}$	O O O	106 109 120	A11 B14 A16	Red, green, and blue complementary analog outputs. These analog outputs are the complement of the red, green, and blue analog outputs. ***When these outputs are not used, they should be terminated to analog GND through resistors that match the impedance on the positive DAC outputs. For example, when the DAC supplies a doubly terminated 75Ω load, the complimentary outputs should be terminated with 37.5Ω resistors. The next best solution is leave these pins unconnected and set the SHUNT bit of the DAC Control Register (register 0x000D) to one.***
DAC Support Pins				
VREF	I	103	D15	Voltage Reference. The voltage reference pin should be connected to 1.235 V and decoupled to DAC GND with a 10 nF ceramic capacitor.
RREF	I	114	B10	Resistor Reference. The resistor reference pin connects to an internal op amp which compares the voltage on this pin to that of the voltage reference, VREF. The current flowing out of the resistor reference, RREF, is such that the voltage developed across the reference resistor matches the voltage reference, VREF. The value of the resistor connected to RREF determines the full scale output current of the DACs. A value of 729Ω is recommended.
CVREF	I	102	C16	Comparator Voltage Reference. This pin is set to 0.35 V by an internal voltage divider between VREF and DAC GND. This level is used internally by comparators to sense the values of the DAC outputs. The CVREF pin should be decoupled to DAC GND with a 1 nF ceramic capacitor.
GREF	I	112	A09	Gate Reference. The gate reference is the output of the DAC op amp which serves as input to gates of the devices connecting DAC V_{DD} to DAC current sources. The GREF pin should be decoupled to DAC V_{DD} with a 1 nF ceramic capacitor.

Table 34. Pin Description (Continued)

Signal	Type	QFP Pin(s)	BGA Pin(s)	Description
PLL Support Pins				
VIDEO PLLC	I	232	U01	Video PLL Loop Filter. This pin should be connected to a parallel combination of a 1.3KΩ resistor and a 680 pF capacitor. Return the parallel resistor and capacitor to VIDEO PLLCR using an 8.2 nF capacitor.
VIDEO PLLCR	I	231	R01	Video PLL Loop Filter Return. This pin should be connected as described above.
AUX PLLC	I	237	W04	Auxiliary PLL Loop Filter. This pin should be connected to a parallel combination of a 1.3KΩ resistor and a 680 pF capacitor. Return the parallel resistor and capacitor to AUX PLLCR using an 8.2 nF capacitor.
AUX PLLCR	I	236	W02	Auxiliary PLL Loop Filter Return. This pin should be connected as described above.
Manufacturing Test Pins				
N_TEST	I	038	T14	Test Mode. The test mode input must be held high for functional operation. It must be terminated to digital V_{DD}, preferably with a 10KΩ resistor.
$\overline{DI2}$ $\overline{RI}$ DIFF_TEST A_C_CLK B_CLK B_C_CLK ARRAY_A_C ARRAY_B	I I I I I I I I	223 125 036 247 224 222 219 037	R03 C11 R14 R04 P02 N02 P03 N11	Test Inputs. These test inputs must be held high for functional operation. They must be terminated to digital V_{DD}, preferably with a 10KΩ resistor.
Power and Ground Pins				
DIGITAL V_{DD}		002, 020, 049, 068, 070, 088, 134, 136, 155, 184, 202, 204, 221, 254, 272	C03, D06, E04, E08, F02, H04, J08, J15, K02, K09, L08, L13, N15, P09, P12, R06, R10, U08, U12, U14	Digital Power.
DIGITAL GND		001, 019, 035, 048, 069, 087, 098, 135, 156, 169, 185, 203, 225, 226, 243, 244, 255	B04, B06, D02, D08, E06, F04, H02, J09, J13, K04, K08, K12, L09, L15, N13, R08, R12, T15, U10, V05, V13	Digital Ground.

Table 34. Pin Description (Continued)

Signal	Type	QFP Pin(s)	BGA Pin(s)	Description
DAC V_{DD}		104, 105, 108, 113, 115, 117, 118, 119	A10, A13, A14, B12, B15, B16, C12, C14, C15, D16, E12, E13, F11, G10	DAC Power.
DAC GND		101, 111, 116, 122	A08, B09, B11, C10, C13, D12, D13, D14, E14, E15, E16, F12, G11, H10	DAC Ground.
VIDEO PLL V_{DD}		235	M05, N04	Video PLL Power.
VIDEO PLL GND		233	L06, N05, P01, R02, T01, T02, U02, V01	Video PLL Ground.
AUX PLL V_{DD}		239	N06, P05	Auxiliary PLL Power.
AUX PLL GND		234	M06, P04, V02, V03, V04, W01, W03, W05	Auxiliary PLL Ground.

11.3 QFP Pin Assignments

Table 35. QFP Signal List by Pin Number

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
001	DIGITAL GND	069	DIGITAL GND	137	PIX(083)	205	PIX(042)
002	DIGITAL V_{DD}	070	DIGITAL V_{DD}	138	PIX(091)	206	PIX(050)
003	PIX(041)	071	PIX(048)	139	PIX(147)	207	PIX(150)
004	PIX(049)	072	PIX(056)	140	PIX(067)	208	PIX(026)
005	PIX(149)	073	PIX(148)	141	PIX(075)	209	PIX(034)
006	PIX(025)	074	PIX(032)	142	PIX(143)	210	PIX(146)
007	PIX(033)	075	PIX(040)	143	RS(0)	211	PIX(014)
008	PIX(061)	076	PIX(144)	144	PIX(051)	212	VID_REF
009	PIX(053)	077	LCLK	145	PIX(151)	213	PIX(154)
010	HSYNC_OUT	078	PIX(016)	146	PIX(059)	214	PIX(018)
011	PIX(153)	079	PIX(152)	147	PIX(043)	215	PIX(158)
012	PIX(017)	080	PIX(024)	148	PIX(155)	216	PIX(006)
013	PIX(157)	081	PIX(156)	149	PIX(035)	217	PIX(010)
014	VSYNC_OUT	082	PIX(008)	150	PIX(027)	218	PIX(002)
015	PIX(009)	083	PIX(000)	151	PIX(019)	219	ARRAY_A_C
016	PIX(001)	084	PIX(060)	152	PIX(011)	220	MON_ID(2)
017	PIX(093)	085	PIX(044)	153	PIX(095)	221	DIGITAL V_{DD}
018	PIX(101)	086	PIX(052)	154	PIX(003)	222	B_C_CLK
019	DIGITAL GND	087	DIGITAL GND	155	DIGITAL V_{DD}	223	$\overline{\text{DI2}}$
020	DIGITAL V_{DD}	088	DIGITAL V_{DD}	156	DIGITAL GND	224	B_CLK
021	PIX(085)	089	PIX(036)	157	PIX(087)	225	DIGITAL GND
022	PIX(077)	090	PIX(028)	158	PIX(055)	226	DIGITAL GND
023	PIX(069)	091	PIX(020)	159	PIX(079)	227	no connect
024	PIX(037)	092	PIX(119)	160	PIX(063)	228	no connect
025	DATA(2)	093	PIX(012)	161	PIX(071)	229	no connect
026	PIX(045)	094	PIX(127)	162	PIX(047)	230	no connect
027	DATA(3)	095	PIX(004)	163	PIX(039)	231	VIDEO PLLCR
028	PIX(029)	096	PIX(103)	164	HSYNC_IN	232	VIDEO PLLC
029	PIX(021)	097	PIX(111)	165	PIX(031)	233	VIDEO PLL GND
030	DATA(0)	098	DIGITAL GND	166	PIX(007)	234	AUX PLL GND
031	DATA(1)	099	AUX_PLL_OUT	167	PIX(023)	235	VIDEO PLL V_{DD}
032	DDOTCLK	100	$\overline{\text{BLANK}}$	168	PIX(015)	236	AUX PLLCR
033	PIX(005)	101	DAC GND	169	DIGITAL GND	237	AUX PLLC
034	no connect	102	CVREF	170	PIX(118)	238	no connect
035	DIGITAL GND	103	VREF	171	PIX(110)	239	AUX PLL V_{DD}
036	DIFF_TEST	104	DAC V_{DD}	172	PIX(126)	240	no connect
037	ARRAY_B	105	DAC V_{DD}	173	PIX(102)	241	no connect
038	N_TEST	106	$\overline{\text{RED}}$	174	SCLK	242	no connect
039	DATA(4)	107	RED	175	PIX(094)	243	DIGITAL GND
040	DATA(5)	108	DAC V_{DD}	176	PIX(086)	244	DIGITAL GND
041	PIX(013)	109	$\overline{\text{GREEN}}$	177	PIX(062)	245	VID_EXT_CLK
042	PIX(124)	110	GREEN	178	PIX(070)	246	$\overline{\text{VID\_EXT\_CLK}}$
043	DATA(6)	111	DAC GND	179	PIX(054)	247	A_C_CLK
044	PIX(108)	112	GREF	180	PIX(078)	248	MON_ID(1)
045	DATA(7)	113	DAC V_{DD}	181	PIX(046)	249	MON_ID(0)
046	PIX(116)	114	RREF	182	PIX(038)	250	N_RESET
047	PIX(092)	115	DAC V_{DD}	183	PIX(030)	251	MON_ID(3)
048	DIGITAL GND	116	DAC GND	184	DIGITAL V_{DD}	252	AUX_REF
049	DIGITAL V_{DD}	117	DAC V_{DD}	185	DIGITAL GND	253	PIX(125)
050	PIX(076)	118	DAC V_{DD}	186	PIX(122)	254	DIGITAL V_{DD}
051	PIX(084)	119	DAC V_{DD}	187	PIX(022)	255	DIGITAL GND
052	PIX(068)	120	$\overline{\text{BLUE}}$	188	PIX(114)	256	PIX(109)
053	PIX(120)	121	BLUE	189	PIX(106)	257	PIX(117)
054	PIX(112)	122	DAC GND	190	VSYNC_IN	258	PIX(121)
055	PIX(104)	123	no connect	191	PIX(130)	259	PIX(113)
056	PIX(128)	124	PIX(123)	192	PIX(090)	260	PIX(105)
057	PIX(096)	125	$\overline{\text{RI}}$	193	PIX(098)	261	PIX(129)
058	$\overline{\text{CE\_RD}}$	126	PIX(131)	194	PIX(134)	262	PIX(089)
059	PIX(132)	127	RS(1)	195	RS(2)	263	PIX(097)
060	$\overline{\text{CE\_WR}}$	128	PIX(159)	196	PIX(082)	264	PIX(133)
061	PIX(100)	129	PIX(107)	197	PIX(074)	265	PIX(145)
062	PIX(088)	130	PIX(139)	198	PIX(142)	266	PIX(081)
063	PIX(140)	131	PIX(135)	199	PIX(138)	267	PIX(073)
064	PIX(080)	132	PIX(115)	200	PIX(066)	268	PIX(141)
065	PIX(136)	133	PIX(099)	201	PIX(058)	269	PIX(137)
066	PIX(064)	134	DIGITAL V_{DD}	202	DIGITAL V_{DD}	270	PIX(065)
067	PIX(072)	135	DIGITAL GND	203	DIGITAL GND	271	PIX(057)
068	DIGITAL V_{DD}	136	DIGITAL V_{DD}	204	DIGITAL V_{DD}	272	DIGITAL V_{DD}

11.4 Top View: BGA Pin Assignment

	W	V	U	T	R	P	N	M	L	K	J	H	G	F	E	D	C	B	A	
01	AUX PLL GND	VID PLL GND	VID PLL C	VID PLL GND	VID PLL CR	VID PLL GND	PIX [006]	PIX [018]	PIX [042]	PIX [050]	PIX [066]	PIX [098]	V SYNC IN	PIX [038]	PIX [046]	PIX [054]	PIX [086]	SCLK		01
02	AUX PLL CR	AUX PLL GND	VID PLL GND	VID PLL GND	VID PLL GND	B CLK	B_C CLK	VID REF	PIX [034]	DIG VDD	PIX [074]	DIG GND	PIX [122]	DIG VDD	PIX [062]	DIG GND	PIX [118]	PIX [015]	PIX [007]	02
03	AUX PLL GND	AUX PLL GND	+VID EXT CLK	-VID EXT CLK	-DI2	AR RAY A_C	PIX [002]	PIX [010]	PIX [026]	PIX [082]	PIX [090]	PIX [022]	PIX [030]	PIX [110]	PIX [102]	PIX [126]	DIG VDD	PIX [039]	H SYNC IN	03
04	AUX PLL C	AUX PLL GND	MON ID [1]	MON ID [0]	A_C CLK	AUX PLL GND	VID PLL VDD	MON ID [2]	PIX [014]	DIG GND	PIX [106]	DIG VDD	PIX [070]	DIG GND	DIG VDD	PIX [031]	PIX [047]	DIG GND	PIX [079]	04
05	AUX PLL GND	DIG GND	MON ID [3]	AUX REF	N_ RE SET	AUX PLL VDD	VID PLL GND	VID PLL VDD	PIX [158]	PIX [058]	RS [2]	PIX [114]	PIX [094]	PIX [023]	PIX [071]	PIX [063]	PIX [087]	PIX [095]	PIX [003]	05
06	PIX [105]	PIX [109]	PIX [125]	PIX [117]	DIG VDD	PIX [142]	AUX PLL VDD	AUX PLL GND	VID PLL GND	PIX [154]	PIX [130]	PIX [078]	PIX [027]	PIX [155]	DIG GND	DIG VDD	PIX [019]	DIG GND	PIX [043]	06
07	PIX [073]	PIX [097]	PIX [089]	PIX [113]	PIX [129]	PIX [121]	PIX [146]	PIX [150]	PIX [138]	PIX [134]	PIX [060]	PIX [055]	RS [0]	PIX [011]	PIX [035]	PIX [059]	PIX [051]	PIX [067]	PIX [075]	07
08	PIX [065]	PIX [057]	DIG VDD	PIX [141]	DIG GND	PIX [145]	PIX [081]	PIX [133]	DIG VDD	DIG GND	DIG VDD	PIX [151]	PIX [147]	PIX [091]	DIG VDD	DIG GND	PIX [083]	PIX [115]	DAC GND	08
09	PIX [049]	PIX [137]	PIX [061]	PIX [149]	PIX [041]	DIG VDD	PIX [025]	PIX [153]	DIG GND	DIG VDD	DIG GND	PIX [131]	PIX [159]	PIX [099]	RS [1]	PIX [143]	PIX [107]	DAC GND	GREF	09
10	PIX [033]	H SYNC OUT	DIG GND	PIX [017]	DIG VDD	PIX [053]	PIX [157]	PIX [132]	PIX [136]	PIX [140]	PIX [144]	DAC GND	DAC VDD	PIX [135]	PIX [139]	PIX [119]	DAC GND	RREF	DAC VDD	10
11	V SYNC OUT	PIX [101]	PIX [009]	PIX [093]	PIX [001]	PIX [037]	AR RAY B	PIX [112]	PIX [104]	PIX [148]	PIX [152]	-BL ANK	DAC GND	DAC VDD	PIX [103]	PIX [123]	-RI	DAC GND	-RED	11
12	PIX [085]	PIX [069]	DIG VDD	PIX [077]	DIG GND	DIG VDD	PIX [128]	PIX [084]	PIX [100]	DIG GND	PIX [016]	PIX [156]	PIX [004]	DAC GND	DAC VDD	DAC GND	DAC VDD	DAC VDD	+RED	12
13	DATA [2]	DIG GND	DATA [3]	PIX [021]	DATA [1]	DATA [4]	DIG GND	-CE RD	DIG VDD	LCLK	DIG GND	PIX [000]	PIX [044]	PIX [127]	DAC VDD	DAC GND	DAC GND	+GRN	DAC VDD	13
14	PIX [029]	PIX [045]	DIG VDD	N TEST	DIFF TEST	DATA [6]	PIX [068]	PIX [088]	PIX [064]	PIX [024]	PIX [048]	PIX [052]	PIX [012]	PIX [028]	DAC GND	DAC GND	DAC VDD	-GRN	DAC VDD	14
15	DATA [0]	DDOT CLK	PIX [005]	DIG GND	PIX [013]	PIX [108]	DIG VDD	PIX [096]	DIG GND	PIX [032]	DIG VDD	PIX [008]	PIX [020]	AUX PLL OUT	DAC GND	VREF	DAC VDD	DAC VDD	+BLU	15
16	PIX [116]	DATA [5]	PIX [124]	DATA [7]	PIX [092]	PIX [076]	PIX [120]	-CE WT	PIX [080]	PIX [072]	PIX [056]	PIX [040]	PIX [036]	PIX [111]	DAC GND	DAC VDD	CV REF	DAC VDD	-BLU	16
	W	V	U	T	R	P	N	M	L	K	J	H	G	F	E	D	C	B	A	

11.5 Bottom View: BGA Pin Assignment

	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	T	U	V	W	
01		SCLK	PIX [086]	PIX [054]	PIX [046]	PIX [038]	V SYNC IN	PIX [098]	PIX [066]	PIX [050]	PIX [042]	PIX [018]	PIX [006]	VID PLL GND	VID PLL CR	VID PLL GND	VID PLL C	VID PLL GND	AUX PLL GND	01
02	PIX [007]	PIX [015]	PIX [118]	DIG GND	PIX [062]	DIG VDD	PIX [122]	DIG GND	PIX [074]	DIG VDD	PIX [034]	VID REF	B_C CLK	B CLK	VID PLL GND	VID PLL GND	VID PLL GND	AUX PLL GND	AUX PLL CR	02
03	H SYNC IN	PIX [039]	DIG VDD	PIX [126]	PIX [102]	PIX [110]	PIX [030]	PIX [022]	PIX [090]	PIX [082]	PIX [026]	PIX [010]	PIX [002]	AR RAY A_C	-DI2	-VID EXT CLK	+VID EXT CLK	AUX PLL GND	AUX PLL GND	03
04	PIX [079]	DIG GND	PIX [047]	PIX [031]	DIG VDD	DIG GND	PIX [070]	DIG VDD	PIX [106]	DIG GND	PIX [014]	MON ID [2]	VID PLL VDD	AUX PLL GND	A_C CLK	MON ID [0]	MON ID [1]	AUX PLL GND	AUX PLL C	04
05	PIX [003]	PIX [095]	PIX [087]	PIX [063]	PIX [071]	PIX [023]	PIX [094]	PIX [114]	RS [2]	PIX [058]	PIX [158]	VID PLL VDD	VID PLL GND	AUX PLL VDD	N_ RE SET	AUX REF	MON ID [3]	DIG GND	AUX PLL GND	05
06	PIX [043]	DIG GND	PIX [019]	DIG VDD	DIG GND	PIX [155]	PIX [027]	PIX [078]	PIX [130]	PIX [154]	VID PLL GND	AUX PLL GND	AUX PLL VDD	PIX [142]	DIG VDD	PIX [117]	PIX [125]	PIX [109]	PIX [105]	06
07	PIX [075]	PIX [067]	PIX [051]	PIX [059]	PIX [035]	PIX [011]	RS [0]	PIX [055]	PIX [060]	PIX [134]	PIX [138]	PIX [150]	PIX [146]	PIX [121]	PIX [129]	PIX [113]	PIX [089]	PIX [097]	PIX [073]	07
08	DAC GND	PIX [115]	PIX [083]	DIG GND	DIG VDD	PIX [091]	PIX [147]	PIX [151]	DIG VDD	DIG GND	DIG VDD	PIX [133]	PIX [081]	PIX [145]	DIG GND	PIX [141]	DIG VDD	PIX [057]	PIX [065]	08
09	GREF	DAC GND	PIX [107]	PIX [143]	RS [1]	PIX [099]	PIX [159]	PIX [131]	DIG GND	DIG VDD	DIG GND	PIX [153]	PIX [025]	DIG VDD	PIX [041]	PIX [149]	PIX [061]	PIX [137]	PIX [049]	09
10	DAC VDD	RREF	DAC GND	PIX [119]	PIX [139]	PIX [135]	DAC VDD	DAC GND	PIX [144]	PIX [140]	PIX [136]	PIX [132]	PIX [157]	PIX [053]	DIG VDD	PIX [017]	DIG GND	H SYNC OUT	PIX [033]	10
11	-RED	DAC GND	-RI	PIX [123]	PIX [103]	DAC VDD	DAC GND	-BL ANK	PIX [152]	PIX [148]	PIX [104]	PIX [112]	AR RAY B	PIX [037]	PIX [001]	PIX [093]	PIX [009]	PIX [101]	V SYNC OUT	11
12	+RED	DAC VDD	DAC VDD	DAC GND	DAC VDD	DAC GND	PIX [004]	PIX [156]	PIX [016]	DIG GND	PIX [100]	PIX [084]	PIX [128]	DIG VDD	DIG GND	PIX [077]	DIG VDD	PIX [069]	PIX [085]	12
13	DAC VDD	+GRN	DAC GND	DAC GND	DAC VDD	PIX [127]	PIX [044]	PIX [000]	DIG GND	LCLK	DIG VDD	-CE RD	DIG GND	DATA [4]	DATA [1]	PIX [021]	DATA [3]	DIG GND	DATA [2]	13
14	DAC VDD	-GRN	DAC VDD	DAC GND	DAC GND	PIX [028]	PIX [012]	PIX [052]	PIX [048]	PIX [024]	PIX [064]	PIX [088]	PIX [068]	DATA [6]	DIFF TEST	N TEST	DIG VDD	PIX [045]	PIX [029]	14
15	+BLU	DAC VDD	DAC VDD	VREF	DAC GND	AUX PLL OUT	PIX [020]	PIX [008]	DIG VDD	PIX [032]	DIG GND	PIX [096]	DIG VDD	PIX [108]	PIX [013]	DIG GND	PIX [005]	DDOT CLK	DATA [0]	15
16	-BLU	DAC VDD	CV REF	DAC VDD	DAC GND	PIX [111]	PIX [036]	PIX [040]	PIX [056]	PIX [072]	PIX [080]	-CE WT	PIX [120]	PIX [076]	PIX [092]	DATA [7]	PIX [124]	DATA [5]	PIX [116]	16
	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	T	U	V	W	

11.6 External Circuitry

External components are required to generate current and voltage references for the analog DAC and PLL circuits, for termination of unused inputs, and to decouple power supplies from noise sources.

11.6.1 Analog Voltages

The DAC and PLL's require that external components be attached as shown in Figure 7 on page 96 and summarized in Table 36. All components are connected to analog supplies, analog V_{DD} or analog GND.

The card analog GND plane should be solid.

The card analog V_{DD} plane should be segmented with separate PLL and DAC supplies. These can be derived from the digital supply through a 1 nH inductor, with separate bulk and high frequency decoupling capacitors. Care should be taken to ensure digital signals do not radiate noise into quiet analog circuits and are not wired over the analog power planes, DAC V_{DD}, video PLL V_{DD} and auxiliary PLL V_{DD}.

The on-chip PLL is very sensitive to card generated noise. For optimal performance, follow guidelines that reduce noise in high performance card designs.

Component values and vendor part numbers are provided for reference, but other devices with similar characteristics are acceptable. All elements should be placed as close to the module pins as possible.

11.6.2 DAC Outputs

Ideally, red, green, and blue analog outputs should be wired on a separate signal plane with adjacent analog GND wires running in parallel on both sides to isolate them from potential digital signal noise.

Digital signals should not be wired near the DAC outputs on any wiring level. DAC outputs are clamped and protected from monitor flash-over on-chip and require no additional component connections.

11.6.3 PLL Components

The PLL components must be carefully placed, especially the filter components. This is necessary to avoid pixel shift and jitter phenomena in the display. The PLL components should be placed on the top side of the card directly adjacent to their appropriate pin connections. Digital signals should not be wired near PLL external components or power planes.

11.6.4 Decoupling

For high frequency decoupling, a 0.1 µF capacitor in parallel with a 0.01 µF capacitor should be placed on all power supply pins, as close to the module as possible. The analog and digital power planes should be decoupled from each other using a 1 nH inductor or ferrite bead. All external components should be placed as close as possible to the module and returned to the appropriate module power supply pin.

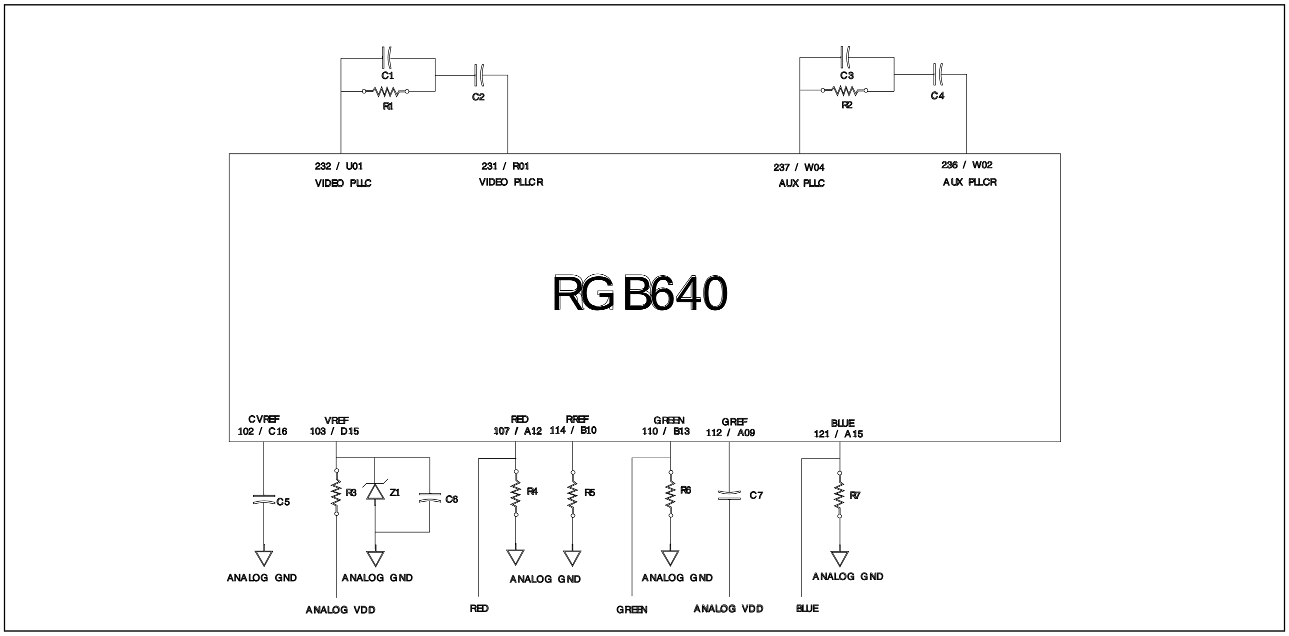


Figure 7. External Components Schematic

Table 36. Component Values

Component	Value	Tolerance	Vendor P/N
R1, R2	1.3 KΩ	5%	Panasonic ERJ3GVYJ132S
R3	1.0 KΩ	5%	Panasonic ERJ3GVYJ102S
R4, R6, R7	75 Ω, 100 Ω	1%	match video cable impedance
R5	729 Ω 938 Ω	1% 1%	for doubly terminated 75 Ω DAC output for doubly terminated 100 Ω DAC output
C1, C3	680 pF 680 pF	10% 5%	Kyocera 1206C681K3B05 Vitramon
C2, C4	8.2 nF	10%	—
C5, C7	0.001 µF	10%	Kyocera 0603X102K2B02
C6	0.01 µF	10%	Kyocera 1206X103K2B02
Z1	1.2 V REF		National Semiconductor LM385-1.2

12.0 Electrical and Timing Specifications

Table 37. Recommended Operating Conditions

Parameter	Symbol	Min	Typ	Max	Units
Power Supply[1]	V_{DD} DAC V_{DD} Video PLL V_{DD} Auxiliary PLL V_{DD}	3.0 3.0 3.0 3.0	3.3 3.3 3.3 3.3	3.8 3.8 3.8 3.8	V
Case Temperature[1]	T_C	0		100	˚C
DAC Output Load	R_L	37.5		50	Ω
Reference Voltage	V_{REF}	1.204		1.266	V

1. For the 220 MHz product (IBM37RGB640CF22), if the video clock is supplied externally using the differential receiver inputs VID_EXT_CLK, $\overline{\text{VID\_EXT\_CLK}}$, there is a restriction on the V_{DD} and T_C operating conditions, given below:

Power Supply	V_{DD}	3.4		3.8	V
Case Temperature	T_C	0		70	˚C

Table 38. Absolute Maximum Ratings

Parameter	Symbol	Min.	Max.	Units
Power Supply	V_{DD} DAC V_{DD} Video PLL V_{DD} Auxiliary PLL V_{DD}	-0.5 -0.5 -0.5 -0.5	5.0 5.0 5.0 5.0	V
Signal Pin Voltage		-0.5	5.5	V
DAC Output Short Circuit Duration	T_{SC}		∞	sec
Case Temperature	T_C	0	145	˚C
Soldering Temperature (5 seconds, 0.25 in. from case)	T_{SOL}		260	˚C
Vapor Phase Soldering Temperature (1 minute)	$T_{V,SOL}$		260	˚C

Table 39. DC Characteristics

Parameter	Symbol	Conditions	Min.	Typ	Max.	Units
CMOS Digital Inputs						
Input High Voltage	V_{IH}		2.0		5.5	V
Input Low Voltage	V_{IL}		-0.5		0.8	V
Input High Current	I_{IH}	V_{IH} = 2.4V			20	µA
Input Low Current	I_{IL}	V_{IL} = 0.0V	-20			µA
Input Capacitance	C_I	f=1 MHz		4	8	pF
ECL Digital Inputs						
ECL High Voltage	V_{EIH}		2.3		2.95	V
ECL Low Voltage	V_{EIL}		1.5		2.1	V
ECL Common Mode	V_{CM}		2.0	2.2	2.4	V
ECL Differential	V_{Δ}		0.3		1.0	V
Differential Input Current	I_{IH}	V_{IN} = 3.3V	0		1	µA
	I_{IL}	V_{IN} = -0.5V	0		-1	µA
Input Capacitance	C_I	f=1 MHz			10	pF
Digital Outputs						
Output High Voltage	V_{OH}	I_{OH} = -12 mA	2.4			V
Output Low Voltage	V_{OL}	I_{OL} = 6 mA			0.4	V
DAC Analog Outputs						
Resolution					10	Bits
Accuracy (10 bit)						
Monotonicity		Guaranteed				
Absolute Full Scale	AFS				±5	%
Integral Linearity Error	ILE				±2 1⁄2	LSB
Differential Linearity Error	DLE				+11⁄2 / -1	LSB
DAC-to-DAC Matching				±1	±3	%
Output Transition						
Fast	t_R	R_L = 37.5 Ω		0.7		ns
	t_F	R_L = 37.5 Ω		2.5		ns
Slow	t_R	R_L = 37.5 Ω		1.2		ns
	t_F	R_L = 37.5 Ω		5.0		ns

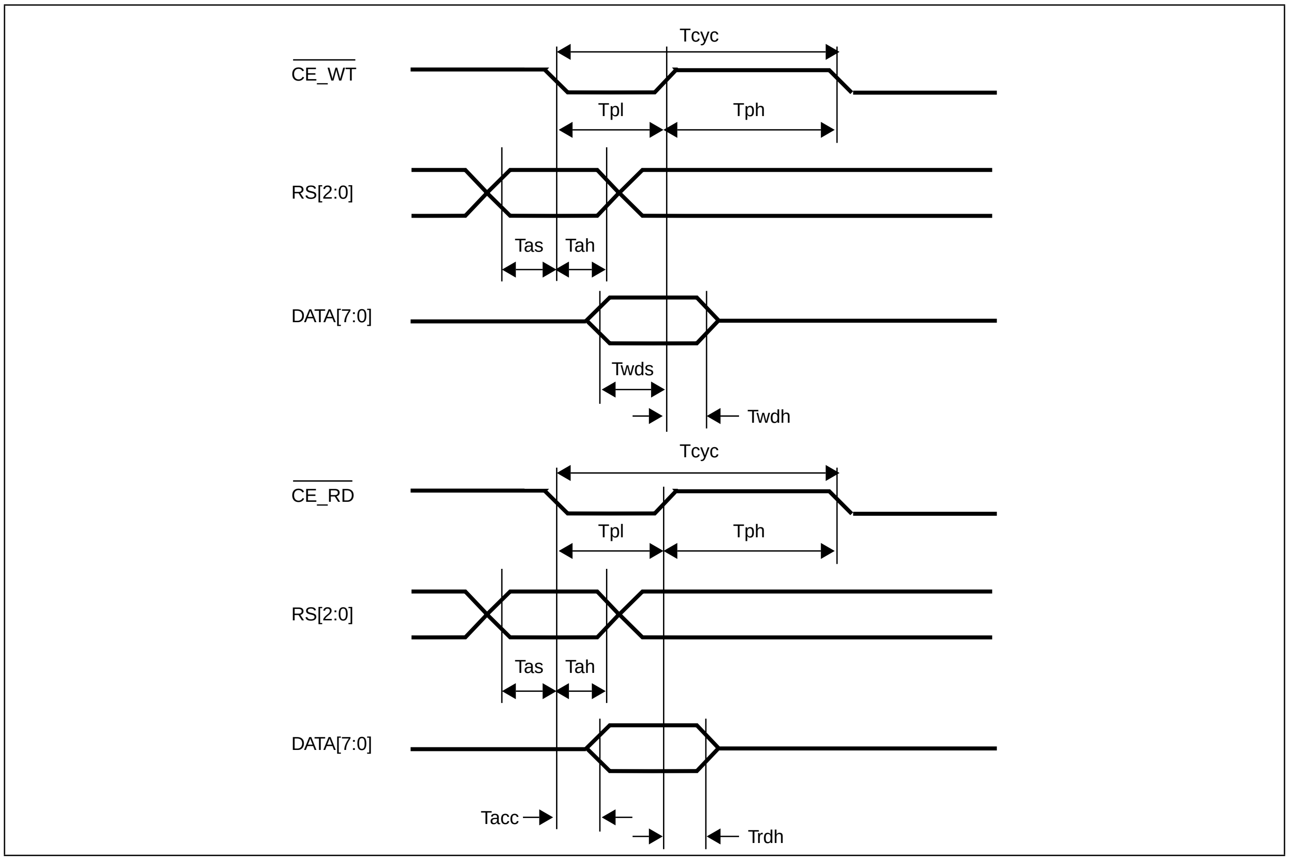


Figure 8. Processor Interface Timings

Table 40. Processor Interface Timings

Symbol	Description	Min (Video Clock < 100 MHz)	Min (Video Clock > 100 MHz)
All Locations			
Tas	Address Setup	5 ns	5 ns
Tah	Address Hold	5 ns	5 ns
Twds	Write Data Setup	5 ns	5 ns
Twdh	Write Data Hold	10 ns	10 ns
Trdh	Read Data Hold	2 ns	2 ns
Palette and Cursor Pixel Map Locations Only			
Tcyc	CE_RD / CE_WT Cycle Time	6 video clocks	60 ns
Tpl	CE_RD / CE_WT Active Time	3 video clocks	30 ns
Tph	CE_RD / CE_WT Restore Time	3 video clocks	30 ns
	Any Non Auto-Increment Read (Figure 9)	7 video clocks	75 ns
Tacc	Read Data Access	2 video clocks	25 ns
All Other Locations			
Tcyc	CE_RD / CE_WT Cycle Time	60 ns	60 ns
Tpl	CE_RD / CE_WT Active Time	30 ns	30 ns
Tph	CE_RD / CE_WT Restore Time	30 ns	30 ns
Tacc	Read Data Access	25 ns	25 ns

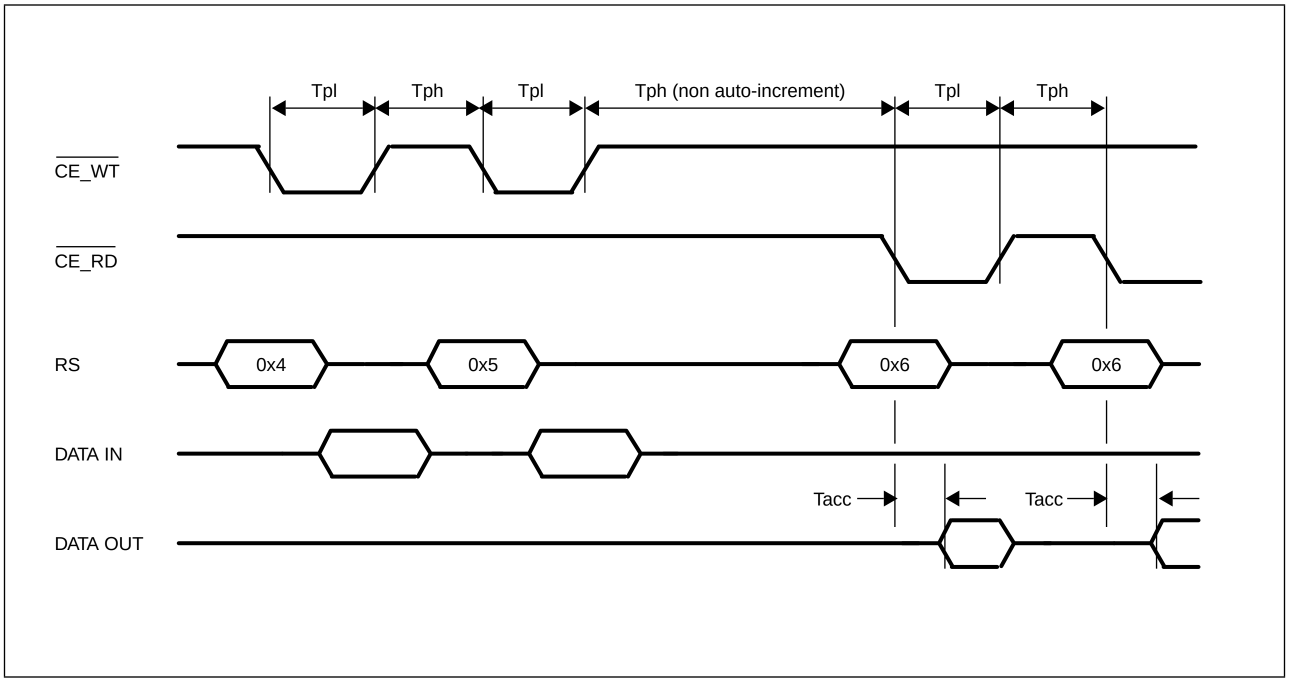
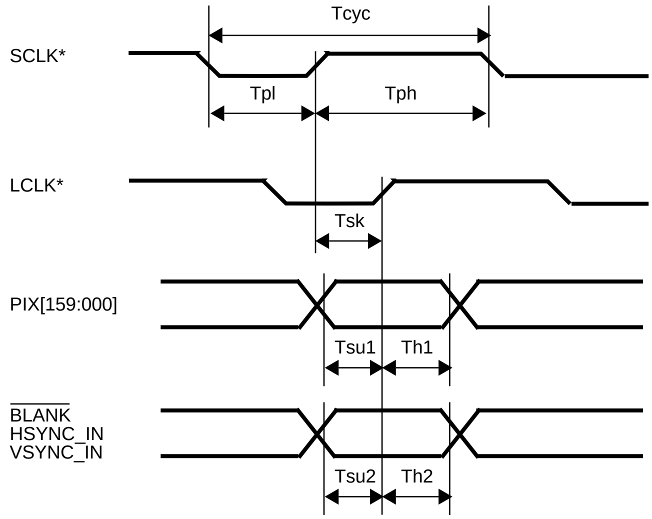


Figure 9. Non Auto-Increment Palette and Cursor Pixel Map Read. Following the write of the index registers, the first read of the palette or cursor pixel map must be delayed to allow for pre-fetching of the RAM contents. Subsequent reads can be faster if the index is auto-incremented.



* See note in Section 7.6 on page 38 stating that for non-VGA mode LCLK must be derived from SCLK

Figure 10. Serial Data Interface Timings

Table 41. Serial Data Interface Timings for 2:1, 4:1, 8:1 and 16:1 Modes

Symbol	Description	Min	Max
Tcyc	Serial Clock Cycle Time	16 ns	—
Tpl	Serial Clock Pulse Lo	8 ns	—
Tph	Serial Clock Pulse Hi	8 ns	—
Tsk	Load Clock Skew	0 ns	Tcyc
Tsu1	Serial Data Setup Time to Load Clock	1 ns	—
Th1	Serial Data Hold Time to Load Clock	3 ns	—
Tsu2	Misc Signal Setup Time to Load Clock	1 ns	—
Th2	Misc Signal Hold Time to Load Clock	3 ns	—

Table 42. Serial Data Interface Timings for 16:3 Mode

Symbol	Description	Min	Max
Tcyc	Serial Clock Cycle Time	16 ns	—
Tpl	Serial Clock Pulse Lo	8 ns	—
Tph	Serial Clock Pulse Hi	8 ns	—
Tsk	Load Clock Skew	0 ns	Tcyc - 8 ns
Tsu1	Serial Data Setup Time to Load Clock	1 ns	—
Th1	Serial Data Hold Time to Load Clock	3 ns	—
Tsu2	Misc Signal Setup Time to Load Clock	1 ns	—
Th2	Misc Signal Hold Time to Load Clock	3 ns	—

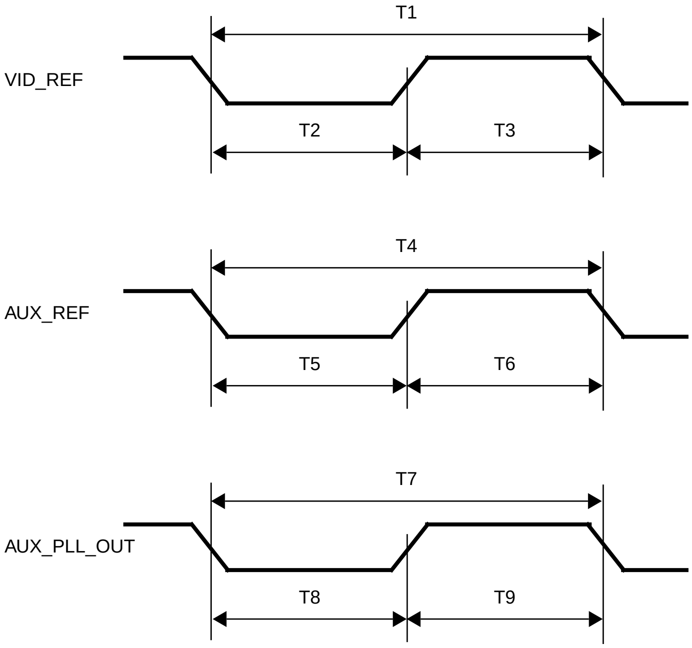


Figure 11. PLL Input and Output Timings

Table 43. Clock Timings

Symbol	Description	Min
T1	Video PLL Reference Cycle Time	10 ns
T2	Video PLL Reference Lo Time	4 ns
T3	Video PLL Reference Lo Time	4 ns
T4	Auxiliary PLL Reference Cycle Time	10 ns
T5	Auxiliary PLL Reference Lo Time	4 ns
T6	Auxiliary PLL Reference Lo Time	4 ns
T7	Auxiliary PLL Output Cycle Time	10 ns
T8	Auxiliary PLL Output Lo Time	4 ns
T9	Auxiliary PLL Output Lo Time	4 ns

RREF = 938 OHMS				RREF = 729 OHMS			
RED, BLUE		GREEN		RED, BLUE		GREEN	
mA	V	mA	V	mA	V	mA	V
13.99	0.698	20.00	1.000	18.65	0.698	26.67	1.000
0.00	0.000	6.01	0.302	0.00	0.000	8.05	0.302
		0.00	0.000			0.00	0.000

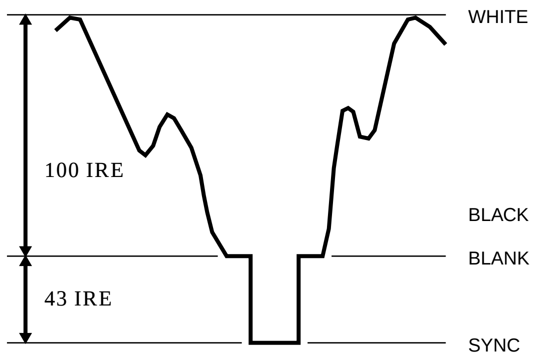


Figure 12. Composite DAC Output (Setup = 0.0 IRE). For 100Ω and 75Ω doubly terminated loads, RS-343A levels, blank pedestal = 0.0 IRE, sync on green.

RREF = 938 OHMS				RREF = 729 OHMS			
RED, BLUE		GREEN		RED, BLUE		GREEN	
mA	V	mA	V	mA	V	mA	V
14.28	0.714	20.00	1.000	19.05	0.714	26.67	1.000
1.04	0.054	6.78	0.340	1.44	0.054	9.05	0.340
0.00	0.000	5.71	0.286	0.00	0.000	7.62	0.286
		0.00	0.000			0.00	0.000

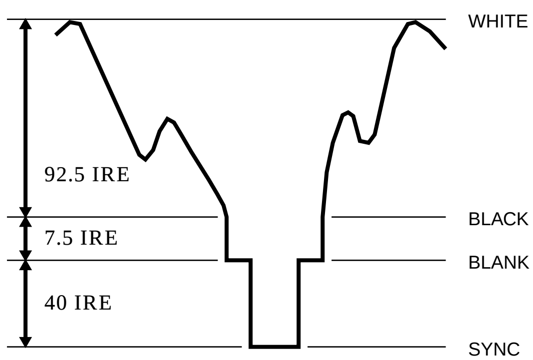
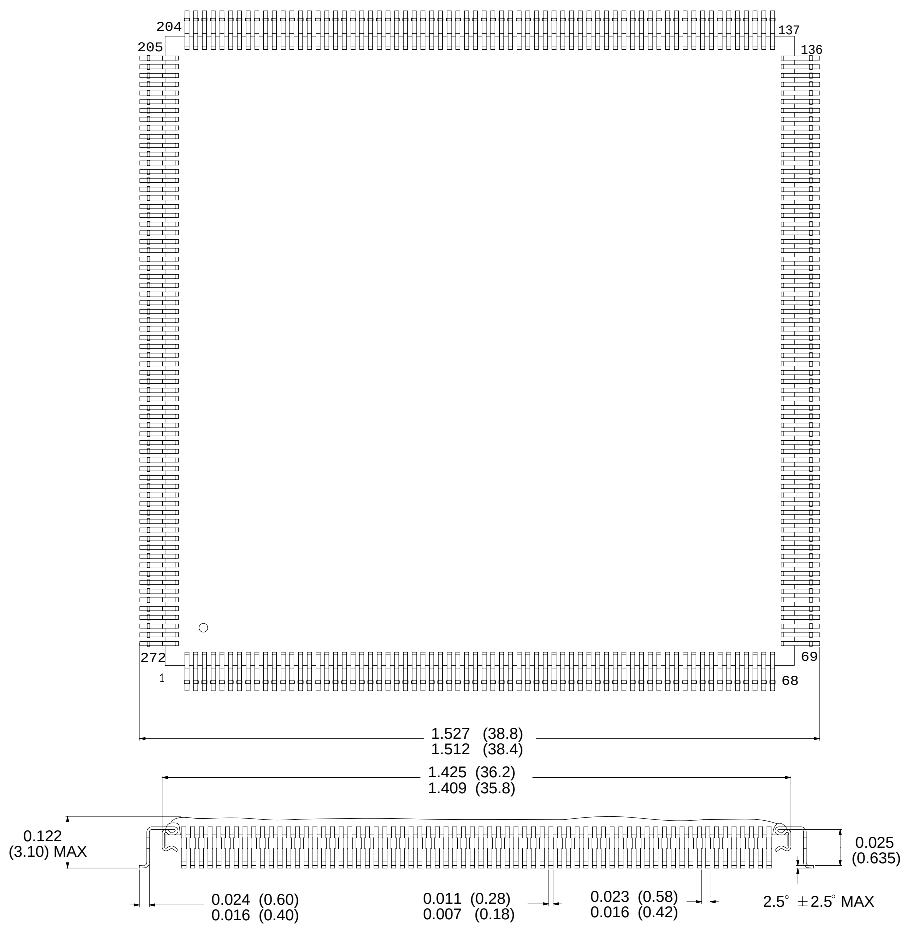


Figure 13. Composite DAC Output (Setup = 7.5 IRE). For 100Ω and 75Ω doubly terminated loads, RS-343A levels, blank pedestal = 7.5 IRE, sync on green.

13.0 Package Information



Top view, heat sink not shown. Drawing not to scale. in(mm)

Figure 14. 272-pin Quad Flat Pack

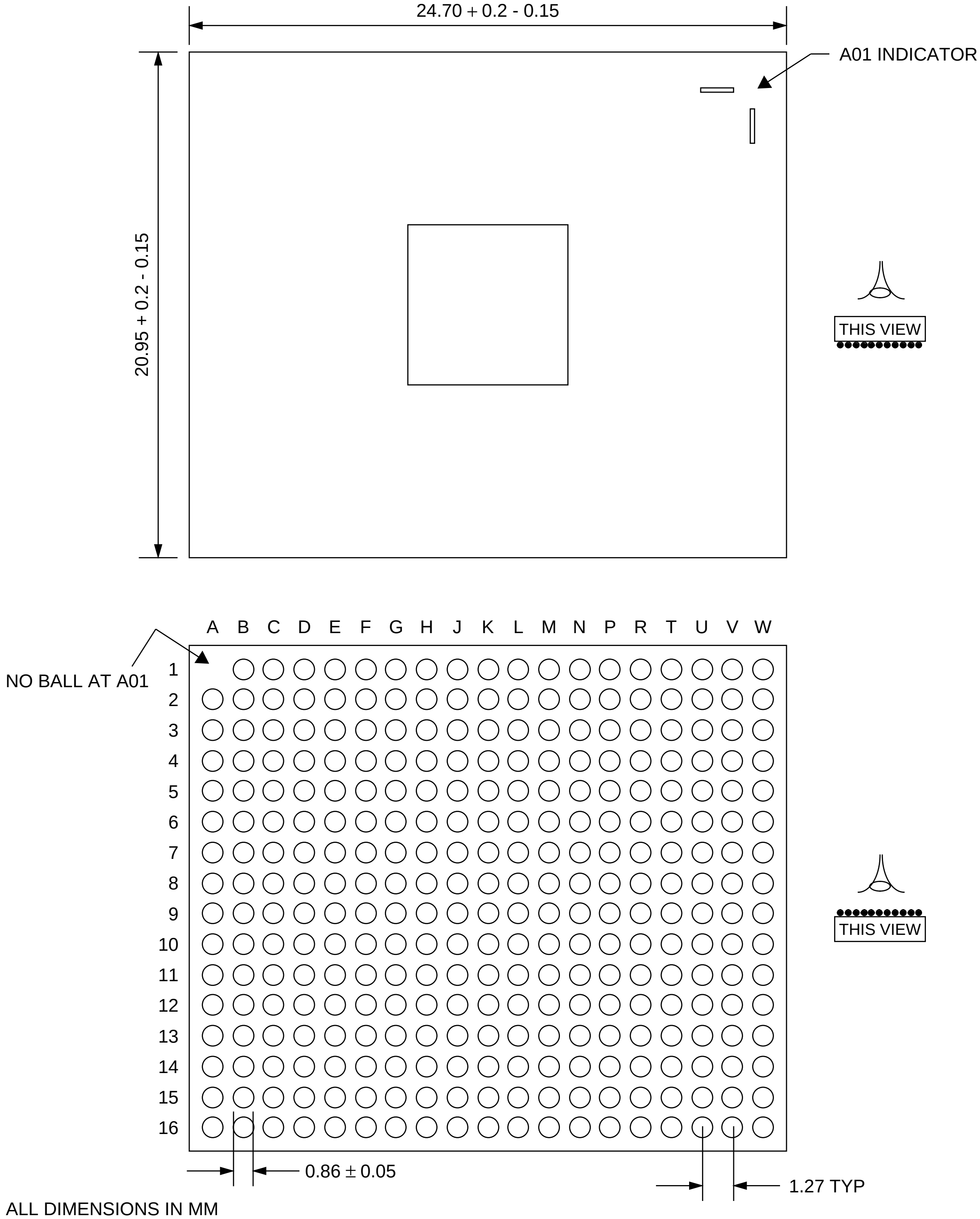


Figure 15. 304-pin Ball Grid Array Package

14.0 Ordering Information

Table 44. Part Numbers

Part Number	Speed	Package
IBM 37RGB640 CF 17	170 MHz	CQFP
IBM 37RGB640 CF 22	220 MHz	CQFP
IBM 37RGB640 CB 17	170 MHz	CBGA
IBM 37RGB640 CB 22	220 MHz	CBGA

15.0 Change Summary

Table 45. Summary of Changes

Date	Changes
06/26/95	1. First publication.
03/25/96	This revision contains a number of formatting and typographic changes. The general organization of the document remains the same, but some sections have been moved. Because of these changes the section, table and figure numbering has changed throughout the document. 1. Section 1.0 is expanded to add introductory material. 2. The description of Load Clock Interleave is moved from the WAT chapter (it was section 3.4) to the Serializer chapter (Section 2.5 on page 8). 3. The WAT chapter (previously 3.0) is divided into two chapters: Pixel Formats (3.0) and Window Attribute Tables (4.0). 4. The new Pixel Formats chapter (3.0) expands the description of YUV to RGB conversion. 5. The section “Chroma Key Operation” (previously 3.3.5) is now a paragraph in the description of bit 3, CKE, of Byte 3 of the Overlay Window Attribute Table (section 4.2.4). 6. The Operations chapter (previously 4.0) is divided into three chapters: Cursor and Crosshair (5.0), VGA (6.0) and Clocking (7.0). The following changes are additions or re-wordings made for clarity: 7. In section 5.1.2, “Processor Access Of The Pixel Map” on page 30, added the paragraphs starting with “To begin a read access … ” and “Subsequent read accesses … “. 8. In sections 6.0, “VGA,” and 7.5, “Serial Clock,” added text noting that the serial clock operation is indeterminate in VGA mode (and therefore user will have to supply a clock to the LCLK input in VGA mode). 9. In section 7.3.1, “Additional Constraints” on page 37, equation (5) is restructured. It is made clear in this section, and in 7.3.2, “Programming Summary,” item 4., that for the 3 MHz guardband on PLL frequencies, the frequencies to compare are the CLOCKOUT frequencies. 10. Added section 7.3.4, “PLL Reset” on page 37. 11. In section 9.4.1.2 , “Reading The Palette” on page 42, added the paragraphs starting with “To begin a read access … ” and “Subsequent read accesses … “. 12. For the DAC Compare / Monitor ID register (page 54), described the mapping of MON_ID pins to the processor data bus during a read operation. 13. For the Sync Control register (page 57), defined the polarity of the CSE bit. 14. For the Auxiliary PLL Reference Divide register (page 61) added text regarding glitch-free transition in frequency. 15. In Table 34, "Pin Description," for the Processor Interface signals $\overline{CE\_RD}$ and $\overline{CE\_WT}$, added a note that these signals should not be allowed to float if connected to a 3-state net. 16. In Table 34, "Pin Description," for the Miscellaneous signal N_RESET, noted that this input is active low.

Table 45. Summary of Changes (Continued)

Date	Changes
03/25/96 (Continued)	The following changes correct errors in the original edition: 17. In section 5.1.6, “Blinking” on page 31, and in the register descriptions for the Cursor Blink Rate (page 65) and Cursor Blink Duty Cycle (page 65) the timing of the blinking rate and duty cycle is corrected. 18. In section 8.1, “MISR,” added a note that the MISR Registers contain the *inverse* of the actual diagnostic signature. Added a similar note to the description of the MISR registers (sections 10.62 through 10.65). 19. In section 9.5, “Cursor Pixel Map Access” on page 42, removed the sentence: “Access to the pixel map is the same as access to any indexed register.” 20. In Table 29, "Single Cycle Addresses," switched the register locations for the Crosshair Horizontal and Vertical Pattern registers. The same change is made for the register descriptions on page 76. (The correct index for the Crosshair Vertical Pattern register is 0x0055, and the correct index for the Crosshair Horizontal Pattern register is 0x0056.) 21. In the descriptions of the Serializer Control registers (headings 10.3 through 10.8), restricted the range of the serial data interface to PIX(127:000) or PIX(159:120) where appropriate. 22. In Table 34, "Pin Description," for the Miscellaneous signals MON_ID(03:00), reworded how to access the value on these inputs. 23. In Table 34, "Pin Description," for the Power and Ground pins, swapped the pin numbers for the Video PLL versus the Auxiliary PLL. 24. Table 39, “DC Characteristics,” on page 98 is updated in its entirety. 25. In Table 40, “Processor Interface Timings,” on page 99, the timings for Tcyc, Tpl and Tph are broken in two groups: “Palette and Cursor Pixel Map Locations Only,” and “All Other Locations.” Tph for “First Palette Access” is changed to “Any Non Auto-Increment Read.” Figure 9, “Non Auto-Increment Palette and Cursor Pixel Map Read” on page 100 is added to illustrate the non auto-increment read timing. The document numbering has changed: 26. The original document number (June 26, 1995) changed from IOG640DSU-01 to SC22-9865-00. This revision (March 25, 1996) becomes document number SC22-9865-01.
09/03/96	1. In the previous edition the description of how updates to the crosshair position registers are processed was incorrect. Section 5.2.2 “Position” on page 32, and the description of the XHRIM bit of the Diagnostics register on page 82, are updated to describe the correct operation. 2. Similarly, the description of how updates to the cursor position registers was incorrect. Furthermore, the update action applies to all cursor control registers from address 0x0040 through 0x004B, not just the cursor position registers. Section 5.1.7 “Update” on page 31 is added, and the description of the CURIM bit of the Diagnostics register on page 82 is updated to describe the correct operation. 3. For the 220 MHz product, when an external video clock is supplied externally using the differential receiver inputs VID_EXT_CLK, $\overline{\text{VID\_EXT\_CLK}}$, the minimum V_{DD} is raised to 3.4V and the maximum T_C is lowered to 70 ˚C. Added a note to Table 37, “Recommended Operating Conditions,” on page 97 stating this. 4. A note was added to section 7.6, “Load Clock” on page 38 stating that in non-VGA mode LCLK must be derived from SCLK. 5. Removed the designation “Preliminary.”

Table 45. Summary of Changes (Continued)

Date	Changes
11/12/97	1. Added BGA package information: a. Added BGA pins to Table 34, “Pin Description,” on page 85. b. Added section 11.4, “Top View: BGA Pin Assignment” on page 93 and section 11.5, “Bottom View: BGA Pin Assignment” on page 94. c. Updated Figure 7, “External Components Schematic” on page 96. d. Added Figure 15, “304-pin Ball Grid Array Package” on page 105. e. Updated Table 44, “Part Numbers,” on page 106. 2. Changed first paragraph in section 5.1.2, “Processor Access Of The Pixel Map” on page 30. Previously it said that the cursor pixel map could be read from at any time. This is changed to say that the cursor must be “off” when reading the cursor pixel map. 3. Added a note to section 6.0, “VGA” on page 33 stating that the DACs must be explicitly enabled following a reset. 4. Updated description of Auxiliary PLL programming update sequence in section 10.21, “0x0014 Auxiliary PLL Reference Divide” on page 61.

IBM Microelectronics Division
1580 Route 52, Bldg. 504
Hopewell Junction, NY
12533-6531

The IBM home page can be found at:
http://www.ibm.com

The IBM Microelectronics Division home page can be found at:
http://www.chips.ibm.com

The IBM Palette DAC web pages can be found at:

http://www.chips.ibm.com/products/display

SC22-9865-03