TVP3020 Data Manual

Video Interface Palette

SLAS080A
June 1994

Texas Instruments

IMPORTANT NOTICE

Texas Instruments Incorporated (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to current specifications in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Please be aware that TI products are not intended for use in life-support appliances, devices, or systems. Use of TI product in such applications requires the written approval of the appropriate TI officer. Certain applications using semiconductor devices may involve potential risks of personal injury, property damage, or loss of life. In order to minimize these risks, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards. Inclusion of TI products in such applications is understood to be fully at the risk of the customer using TI devices or systems.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

SLAS080, October 1993

Printed in the U.S.A.

Contents

Section *Title* *Page*

Contents (Continued)

List of Illustrations

List of Tables

1 Introduction

The TVP3020 Viewpoint Palette is an advanced Video Interface Palette (VIP) from Texas Instruments implemented in EPIC™ 0.8-micron CMOS process. Maximum flexibility is provided by the pixel multiplexing scheme. The scheme accommodates 64-, 32-, 16-, 8-, and 4-bit pixel buses without any circuit modification. This enables the system to be easily reconfigured for varying amounts of available video RAM. The device supports selection of little- or big-endian data format for the pixel-bus-frame buffer interface. Data can be split into 1, 2, 4, or 8 bit planes for pseudo-color mode or split into 12-, 16- or 24-bit true-color and direct-color modes. For the 24-bit direct color modes, an 8-bit overlay plane is available. The 16-bit direct- and true-color modes can be configured to IBM XGA® (5, 6, 5), TARGA® (5, 5, 5, 1), or (6, 6, 4) as another existing format. An additional 12-bit mode (4, 4, 4, 4) is supported with 4 bits for each color and overlay. An on-chip, IBM XGA-compatible hardware cursor is incorporated so that further increases in graphics system performance are possible. The device is also software compatible with the IMSG176/8 and Bt476/8 color palettes.

An internal frequency doubler is incorporated, allowing convenient and cost-effective clock source alternatives to be utilized.

An auxiliary windowing function and a pixel-port-select function are provided so that overlay or VGA graphics can be displayed on top of direct color inside or outside a specified auxiliary window. Color-keyed switching of direct color and overlay is also supported.

Clocking is provided through one of three inputs (2 TTL- and 1 ECL/TTL-compatible) and is software selectable. The video, shift clock, and reference clock outputs provide a software-selected divide ratio of the chosen clock input.

The TVP3020 has three 256-by-8 color lookup tables with triple 8-bit video digital-to-analog converters (DACs) capable of directly driving a doubly terminated 75-Ω line. The lookup tables are designed with a dual-ported RAM architecture that enables ultra-high speed operation. Sync generation is incorporated on the green output channel. Horizontal sync and vertical sync are fed through the device and optionally inverted to indicate screen resolution to the monitor. A palette-page register is used to provide the additional bits of palette address when 1, 2, or 4 bit planes are used. This allows the screen colors to be changed with only one microprocessor interface unit (MPU) write cycle.

The device features a separate VGA bus that allows data from the feature connector of most VGA-supported personal computers to be fed directly into the palette without the need for external data multiplexing. This allows a replacement graphics board to remain downwards compatible by utilizing the existing graphics circuitry often located on the motherboard.

The Viewpoint VIP is highly system integrated. It can be connected to the serial port of VRAM devices without external buffer logic and connected to many graphics engines directly. The split shift-register transfer function, which is supported by VRAM, is also supported by the TVP3020.

The system-integration concept is carried to manufacturing test and field diagnosis. To support these, several highly integrated test functions have been designed to enable simplified testing of the palette, the graphics board, and the graphics system.

NOTE: The TVP3020 includes circuits that are patented and circuit designs that have patents pending.

EPIC is a trademark of Texas Instruments Incorporated.
XGA is a registered trademark of IBM.
TARGA is a registered trademark of Truevision Incorporated.

1.1 Features

- Second-Generation Video Interface Palette
- Supports System Resolutions of:
  - 1600 × 1280 × 1, 2, 4, 8, 16, 24 Bits/Pixel @ 60-Hz and 72-Hz Refresh Rate
  - 1280 × 1024 × 1, 2, 4, 8, 16, 24 Bits/Pixel @ 60-Hz and 72-Hz Refresh Rate
  - 1024 × 768 × 1, 2, 4, 8, 16, 24 Bits/Pixel @ 60-Hz and 72-Hz Refresh Rate
  - And lower resolutions
- Direct-Color Modes:
  - 24-Bit/Pixel with 8-Bit Overlay
  - 16-Bit/Pixel (5, 6, 5) XGA Configuration
  - 16-Bit/Pixel (6, 6, 4) Configuration
  - 15-Bit/Pixel With 1 Bit Overlay (5, 5, 5, 1) TARGA Configuration
  - 12-Bit/Pixel With 4 Bit Overlay (4, 4, 4, 4)
- True-Color Modes:
  - 24-Bit/Pixel With Gamma Correction
  - 16-Bit/Pixel (5, 6, 5) XGA Configuration With Gamma Correction
  - 16-Bit/Pixel (6, 6, 4) Configuration with Gamma Correction
  - 15-Bit/Pixel (5, 5, 5) TARGA Configuration With Gamma Correction
  - 12-Bit/Pixel (4, 4, 4) With Gamma Correction
- RCLK/SCLK/LCLK Data Latching Mechanism Allows Flexible Control of VRAM Timing
- Direct Interfacing to Video RAM
- Support for Split Shift Register Transfers
- 64-Bit-Wide Pixel Bus
- On-Chip Hardware Cursor:
  - 64 × 64 × 2 Cursor (XGA Functionally Compatible)
  - Full-Window Crosshair
  - Dual-Cursor Mode
- 135-, 175-, and 200-MHz Versions
- Supports Overscan for Creation of Custom Screen Borders
- Versatile Pixel Bus Interface Supports Little- and Big-Endian Data Formats
- Windowed Overlay and VGA Capability
- Color-Keyed Switching of Direct Color and Overlay
- On-Chip Clock Selection
- Internal Frequency Doubler
- Triple 8-Bit D/A Converters
- Analog Output Comparators
- Triple 256 × 8 Color Palette RAMs
- RS-343A Compatible Outputs
- Direct VGA Pass-Through Capability
- Palette Page Register
- Horizontal Zooming Capability
- Software Downward Compatible With IMSG176/8 and Bt476/8
- Directly Interfaces to Graphics Processors
- EPIC 0.8-µm CMOS Process

1.2 Functional Block Diagram

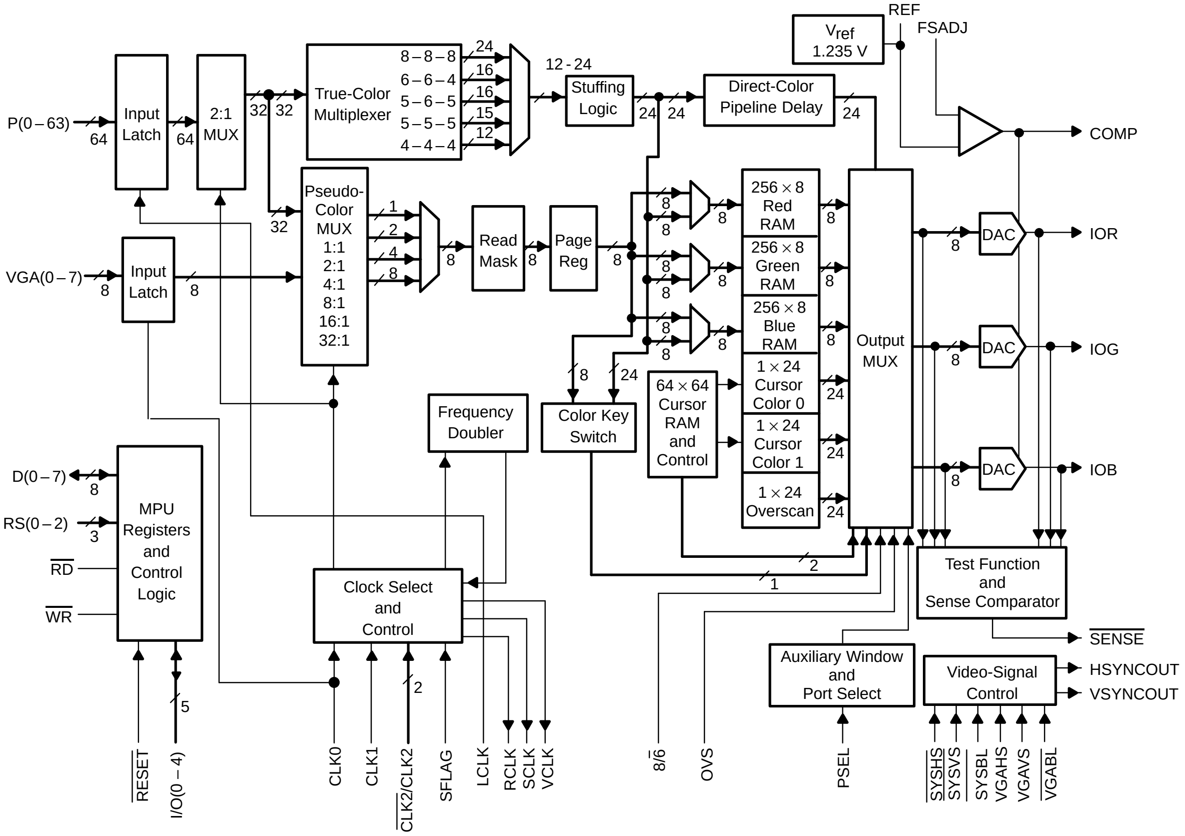
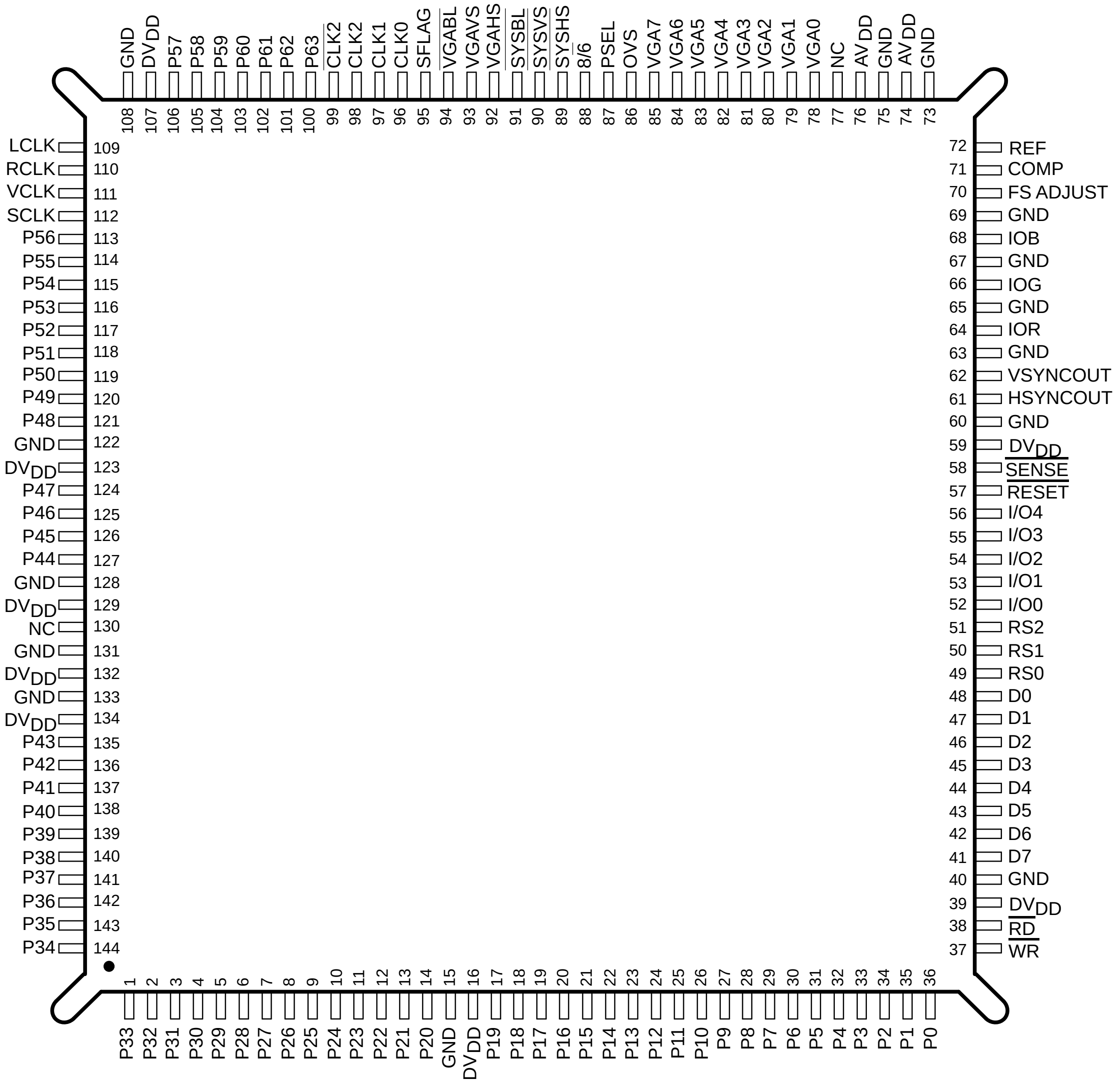


Figure 1–1. Functional Block Diagram

1.3 Terminal Assignments



NC – No internal connection

Figure 1–2. Terminal Assignments

1.4 Ordering Information

TVP3020 – XXX XXX

Pixel Clock Frequency Indicator

MUST CONTAIN THREE CHARACTERS:

–135:135-MHz pixel clock
–175:175-MHz pixel clock
–200:200-MHz pixel clock

Package

MUST CONTAIN THREE LETTERS:

MDN: Metal, Quad Flat Pack
PCE: Plastic, Quad Flat Pack (mechanical data unavailable at time of print)

1.5 Terminal Functions

PIN		I/O	DESCRIPTION
NAME	NO.		
AV_{DD}	74, 76		Analog power. All AV_{DD} terminals must be connected.
CLK0	96	I (TTL compatible)	Dot clock 0 input. CLK0 can be selected to drive the dot clock at frequencies up to 140 MHz. When VGA mode is active, the default clock source is CLK0. The maximum frequency in VGA mode is 85 MHz.
CLK1	97	I (TTL compatible)	Dot clock 1 input. CLK1 can be selected to drive the dot clock at frequencies up to 140 MHz.
CLK2, $\overline{\text{CLK2}}$	98, 99	I (TTL/ECL compatible)	Dual-mode dot clock input. These inputs are essentially ECL-compatible inputs, but two TTL clocks may be used on the CLK2 and $\overline{\text{CLK2}}$ if so selected in the input clock select register. These inputs may be selected as the dot clock up to the device limit while in the ECL mode or up to 140 MHz in the TTL mode.
COMP	71	I	Compensation. COMP provides compensation for the internal reference amplifier. A 0.1-µF ceramic capacitor is required between this terminal and AV_{DD}. The COMP capacitor must be as close to the device as possible to avoid noise pick up.
DV_{DD}	16, 39,59, 107, 123, 129, 132, 134		Digital power. All DV_{DD} terminals must be connected.
D0–D7	41–48	I/O (TTL compatible)	MPU interface data bus. These terminals are used to transfer data in and out of the register map and palette/overlay RAM.
FS ADJUST	70	I	Full-scale adjustment. A resistor connected between this terminal and ground controls the full-scale range of the DACs.
GND	15, 40, 60, 63, 65, 67, 69, 73, 75, 108, 122, 128, 131		Ground. All GND terminals must be connected. The GNDs are connected internally.

NOTE: All unused inputs should be tied to a logic level and not be allowed to float.

1.5 Terminal Functions (Continued)

PIN		I/O	DESCRIPTION
NAME	NO.		
HSYNCOUT	61	O (TTL compatible)	Horizontal sync output after pipeline delay. For system mode the output can be programmed, but for the VGA mode the output carries the same polarity as the input.
IOR, IOG, IOB	64, 66, 68	O	Analog current outputs. These outputs can drive a 37.5-Ω load directly (doubly terminated 75-Ω line), thus eliminating the requirement for any external buffering.
I/O0 – I/O4	52 – 56	I/O (TTL compatible)	Software programmable I/O terminals that can be used to control external devices.
LCLK	109	I (TTL compatible)	Pixel-port-latch clock input. LCLK is used to latch pixel-bus-input data.
OVS	86	I (TTL compatible)	Overscan input. OVS is used to create custom screen borders.
PSEL	87	I (TTL compatible)	Port select. PSEL provides the capability of VGA or overlay windows in a direct color background on a pixel-by-pixel basis.
P0 – P63	1 – 14, 17 – 36, 100 – 106, 113 – 121, 124 – 127, 135 – 144	I (TTL compatible)	Pixel input port. The port can be used in various modes as shown in the multiplexer control register. All the unused terminals need to be tied to GND.
RCLK	110	O (TTL compatible)	Reference clock output. RCLK is essentially the same as the SCLK output but not gated off during blank. It can be used for pixel-port timing reference or other system synchronization.
REF	72		Voltage reference for DACs. An internal voltage reference of nominally 1.235 V is provided, which requires an external 0.1-µF ceramic capacitor between REF and analog GND. However, the internal reference voltage can be overdriven by an externally supplied reference voltage. Typical connection is shown in Appendix A.
$\overline{\text{RESET}}$	57	I	Chip reset. All the registers default to the VGA mode after reset.
$\overline{\text{RD}}$	38	I (TTL compatible)	Read strobe input. A logic 0 on this terminal initiates a read from the register map. Reads are performed asynchronously and are initiated on the low-going edge of $\overline{\text{RD}}$ (see Figure 3–1).
RS0 – RS2	49 – 51	I (TTL compatible)	Register select inputs. These terminals specify the location in the register map that is to be accessed (see Table 2–1).
SCLK	112	O (TTL compatible)	Shift clock output. SCLK is selected as a division of the dot clock input. The output signals are gated off during blank, although SCLK is still used internally to synchronize with the activation of $\overline{\text{BLANK}}$.
$\overline{\text{SENSE}}$	58	O (TTL compatible)	Test mode DAC comparator output signal. This terminal is low if one or more of the DAC output analog levels is above the internal comparator reference of 350 mV ±50 mV.

NOTE: All unused inputs should be tied to a logic level and not be allowed to float.

1.5 Terminal Functions (Continued)

PIN		I/O	DESCRIPTION
NAME	NO.		
SFLAG	95	I (TTL compatible)	Split shift register transfer flag. The Viewpoint VIP detects a low-to-high transition on this terminal during a blank sequence and immediately generates an SCLK pulse. This early SCLK pulse replaces the first SCLK pulse in the normal sequence.
$\overline{\mathrm{SYSBL}}$	91	I (TTL compatible)	System blank input. $\overline{\mathrm{SYSBL}}$ is active (low).
$\overline{\mathrm{SYSHS}}$, $\overline{\mathrm{SYSVS}}$	89, 90	I (TTL compatible)	System horizontal and vertical sync inputs. These signals are used to generate the sync level on the green current output. They are active (low) inputs, but the HSYNCOUT and VSYNCOUT outputs can be programmed through general control register.
VCLK	111	O (TTL compatible)	Video clock output. VCLK is the user-programmable output for synchronization to graphics processor.
$\overline{\mathrm{VGABL}}$	94	I (TTL capability)	VGA blank input. $\overline{\mathrm{VGABL}}$ is active (low).
VGAHS, VGAVS	92, 93	I (TTL capability)	VGA horizontal and vertical sync inputs. These signals are used to generate the sync level on the green current output. They can be either polarity, and the VIP passes the polarities to HSYNCOUT and VSYNCOUT without change.
VGA0 – VGA7	78 – 85	I (TTL capability)	VGA pass-through bus. These buses can be selected as the pixel bus for VGA mode, but it does not allow for any multiplexing.
VSYNCOUT	62	O (TTL capability)	Vertical sync output after pipeline delay. For system mode, the output can be programmed but for the VGA mode, the output carries the same polarity as the input.
$\overline{\mathrm{WR}}$	37	I (TTL capability)	Write strobe input. A logic 0 on this terminal initiates a write to the register map. As with $\overline{\mathrm{RD}}$, write transfers are asynchronous and initiated on the low-going edge of $\overline{\mathrm{WR}}$, (see Figure 3–1).
$8/\overline{6}$	88	I (TTL capability)	DAC resolution selection. This terminal is used to select the data bus width (8 or 6 bits) for the DAC and is essentially provided in order to maintain compatibility with the IMSG176. When this terminal is a logical 1, 8-bit bus transfers are used with D7 the MSB and D0 the LSB. For 6-bit bus operation, while the color palette still has the 8-bit information D5 shifts to the bit 7 position with D0 shifted to the bit 2 position and the two LSBs are filled with zeros at the output multiplexer to DAC. The palette holding register zeroes the two MSBs when it is read in the 6-bit mode.

NOTE: All unused inputs should be tied to a logic level and not be allowed to float.

2 Detailed Description

2.1 MPU Interface

The processor interface is controlled via read and write strobes ($\overline{RD}$, $\overline{WR}$), three register select terminals (RS0 – RS2), and the 8/$\overline{6}$-select terminal. The 8/$\overline{6}$ terminal is used to select between an 8- or 6-bit-wide data path to the color palette RAM and is provided in order to maintain compatibility with the IMSG176. With 8/$\overline{6}$ held low, data on the lowest six bits of the data bus are internally shifted up by two bits to occupy the upper six bits at the output multiplexer and the bottom two bits are then zeroed. This operation is carried out in order to utilize the maximum range of the DACs.

The direct register map is shown in Table 2–1. Extended registers can be accessed through the index register. The index register map is shown in Table 2–2. In general, the index register must first be loaded with the target address value. Successive reads or writes from and to the data register then access the target location. The MPU interface operates asynchronously, with data transfers being synchronized by internal logic.

Table 2–1. Direct Register Map

RS2	RS1	RS0	REGISTER ADDRESSED BY MPU	R/W	DEFAULT (HEX)
0	0	0	Palette Address Register – Write Mode	R/W	XX
0	0	1	Color Palette Holding Register	R/W	XX
0	1	0	Pixel Read Mask	R/W	FF
0	1	1	Palette Address Register – Read Mode	R/W	XX
1	0	0	Reserved		XX
1	0	1	Reserved		XX
1	1	0	Index Register	R/W	XX
1	1	1	Data Register	R/W	XX

Table 2–2. Indirect Register Map (Extended Registers)

INDEX REGISTER (HEX)	R/W	DEFAULT (HEX)	REGISTER ADDRESSED BY INDEX REGISTER
00	R/W	00	Cursor-Position X LSB
01	R/W	00	Cursor-Position X MSB
02	R/W	00	Cursor-Position Y LSB
03	R/W	00	Cursor-Position Y MSB
04	R/W	1F	Sprite-Origin X
05	R/W	1F	Sprite-Origin Y
06	R/W	00	Cursor-Control Register
07			Reserved
08	W	XX	Cursor-RAM Address LSB
09	W	XX	Cursor-RAM Address MSB
0A	R/W	XX	Cursor-RAM Data
0B			Reserved
0C – 0F			Reserved-Undefined
10	R/W	XX	Window-Start X LSB

NOTE: Reserved registers should be avoided; otherwise, circuit behavior could deviate from that specified. Reserved-undefined registers are nonexistent locations on the register map.

Table 2–2. Indirect Register Map (Extended Registers) (Continued)

INDEX REGISTER (HEX)	R/W	DEFAULT (HEX)	REGISTER ADDRESSED BY INDEX REGISTER
11	R/W	XX	Window-Start X MSB
12	R/W	XX	Window-Stop X LSB
13	R/W	XX	Window-Stop X MSB
14	R/W	XX	Window-Start Y LSB
15	R/W	XX	Window-Start Y MSB
16	R/W	XX	Window-Stop Y LSB
17	R/W	XX	Window-Stop Y MSB
18	R/W	80	Multiplexer-Control Register 1
19	R/W	98	Multiplexer-Control Register 2
1A	R/W	00	Input-Clock-Selection Register
1B	R/W	3E	Output-Clock-Selection Register
1C	R/W	00	Palette-Page Register
1D	R/W	20	General-Control Register
1E – 1F			Reserved-Undefined
20	R/W	XX	Overscan Color Red
21	R/W	XX	Overscan Color Green
22	R/W	XX	Overscan Color Blue
23	R/W	XX	Cursor Color 0, Red
24	R/W	XX	Cursor Color 0, Green
25	R/W	XX	Cursor Color 0, Blue
26	R/W	XX	Cursor Color 1, Red
27	R/W	XX	Cursor Color 1, Green
28	R/W	XX	Cursor Color 1, Blue
29	R/W	09	Auxiliary Control Register
2A	R/W	00	General-Purpose I/O Control Register
2B	R/W	XX	General-Purpose I/O Data Register
2C			Reserved
2D			Reserved
2E			Reserved
2F			Reserved
30	R/W	XX	Color-Key OL/VGA Low
31	R/W	XX	Color-Key OL/VGA High
32	R/W	XX	Color-Key Red Low
33	R/W	XX	Color-Key Red High
34	R/W	XX	Color-Key Green Low
35	R/W	XX	Color-Key Green High

NOTE: Reserved registers should be avoided; otherwise, circuit behavior could deviate from that specified. Reserved-undefined registers are nonexistent locations on the register map.

Table 2–2. Indirect Register Map (Extended Registers) (Continued)

INDEX REGISTER (HEX)	R/W	DEFAULT (HEX)	REGISTER ADDRESSED BY INDEX REGISTER
36	R/W	XX	Color-Key Blue Low
37	R/W	XX	Color-Key Blue High
38	R/W	10	Color-Key Control Register
39			Reserved-Undefined
3A	R/W	00	Sense Test Register
3B	R	XX	Test Data Register
3C	R	XX	CRC LSB
3D	R	XX	CRC MSB
3E	W	XX	CRC Control Register
3F	R	20	ID Register
FF	W	XX	Reset Register

NOTE: Reserved registers should be avoided; otherwise, circuit behavior could deviate from that specified. Reserved-undefined registers are nonexistent locations on the register map.

2.2 Color Palette

The color palette is addressed by an internal 8-bit address register for reading/writing data from/to the RAM. This register is automatically incremented following a RAM transfer, allowing the entire palette to be read/written with only one access of the address register. When the address register increments beyond the last location in RAM, it is reset to the first location (address 0). All read and write accesses to the RAM are asynchronous to SCLK, VCLK, and dot clock but performed within one dot clock. Therefore, they do not cause any noticeable disturbance on the display.

The color RAM is 24 bits wide for each location and 8 bits wide for each color. Since the MPU access is eight bits wide, the color data stored in the palette is eight bits even when the six-bit mode is chosen (8/$\overline{6}$ = 0). If the six-bit mode is chosen, the two MSBs of color data in the palette have the values previously written. However, if they are read back in the six-bit mode, the two MSBs are 0s to be compatible with IMSG176 and Bt176. The output multiplexer shifts the six LSB bits to the six MSB positions and fills the two LSBs with 0s after the color palette. The multiplexer then feeds the data to the DAC. The test register and the CRC calculation both take data after the output multiplexer, enabling total system verification. The color palette access is described in the following two sections, and it is fully compatible with IMSG176/8 and Bt476/8.

2.2.1 Writing to Color-Palette RAM

To load the color palette, the MPU must first write to the address register (write mode) with the address where the modification is to start. This is then followed by three successive writes to the palette holding register with 8 bits of red, green, and blue data. After the blue write cycle, the three bytes of color data are concatenated into a 24-bit word that is then written to the RAM location specified by the address register. The address register then increments to the next location, which the MPU may modify by simply writing another sequence of red, green, and blue data. A block of color values in consecutive locations may be written to by writing the start address and performing continuous red, green, and blue write cycles until the entire block has been written.

2.2.2 Reading From Color-Palette RAM

Reading from the palette is performed by writing to the address register (read mode) with the location to be read. This then initiates a transfer from the palette RAM into the holding register, followed by an increment of the address register. Three successive MPU reads from the holding register produce red, green, and blue color data (6 or 8 bits depending on the 8/$\overline{6}$ mode) for the specified location. Following the blue read cycle, the contents of the color-palette RAM at the address specified by the address register are copied into the holding register and the address register is again incremented. As with writing to the palette, a block of color values in consecutive locations may be read by writing the start address and performing continuous red, green, and blue read cycles until the entire block has been read. Since the color-palette RAM is dual ported, the RAM may be read during active display without disturbing the video.

2.2.3 Palette-Page Register

The palette-page register appears as an 8-bit register on the extended register map (see Section 2.1). Its purpose is to provide high-speed color changing by removing the need for palette reloading. When using 1, 2, or 4 bit planes, the additional planes are provided from the page register. When using four bit planes, the pixel inputs specify the lower four bits of the palette address with the upper four bits specified from the page register. This gives the user the capability of selecting from 16 palette pages with only one chip access, thus allowing all the screen colors to be changed at the line frequency. A bit-to-bit correspondence is used; therefore, in the above configuration, page register bits 7 through 4 map onto palette address bits 7 through 4, respectively. This is illustrated in Table 2–3.

> **NOTE:**The additional bits from the page register are inserted after the read mask.
>
> The palette-page register specifies the additional bit planes for the overlay field in direct-color modes with less than 8 bits per pixel overlay.

Table 2–3. Allocation of Palette-Page Register Bits

NUMBER OF BIT PLANES	MSB	PALETTE ADDRESS BITS						LSB
8	M	M	M	M	M	M	M	M
4	P7	P6	P5	P4	M	M	M	M
2	P7	P6	P5	P4	P3	P2	M	M
1	P7	P6	P5	P4	P3	P2	P1	M

Pn = n bit from page register
M = bit from pixel port

2.2.4 Read Masking

The read-mask register is an 8-bit register used to enable or disable a bit plane from addressing the color-palette RAM in the pseudo-color modes. Each palette address bit is logically ANDed with the corresponding bit from the read mask register before going to the palette-page register and addressing the palette RAM.

In order to provide maximum flexibility to control palette data, the read mask operation is performed before the addition of the page register bits. Therefore, care must be taken in those modes that have less than eight bits per pixel of pseudo color or overlay data. Be aware of the palette-page register settings in these modes.

2.3 Clock-Selection and Output-Clock (SCLK, RCLK, and VCLK) Generation

The Viewpoint VIP provides a maximum of four clock inputs. Two are dedicated as TTL inputs; the other two can be selected as either one differential ECL input or two extra TTL inputs. The TTL inputs can be used for video rates up to 140 MHz. The dual-mode clock input (ECL/TTL) is primarily an ECL input but can be used as TTL-compatible inputs if the input-clock-selection register is so programmed. The clock source used at power up is CLK0; an alternative source can be selected by software during normal operation. This chosen clock input can be used unmodified as the dot clock (representing pixel rate to the monitor). Alternatively, if the input-clock-selection register is programmed to use the internal frequency doubler, the chosen clock source is used as a reference for multiplication. The device also allows for user programming of RCLK, SCLK and VCLK outputs (reference, shift and video clocks) by using the output-clock-selection register. The input-clock- and output-clock-selection registers are shown in Tables 2–4 and 2–5 and are located in the indirect register map (see Table 2–2).

The ECL input can be used as a differential or single-ended input. If CLK2 is used as a single-ended ECL input, $\overline{\text{CLK2}}$ needs to be externally terminated to set the input common-mode signal level. This can be done with a simple resistor divided, as is the case with fully differential ECL. Care needs to be taken when choosing the resistor values to ensure that the dc level on $\overline{\text{CLK2}}$ is in the middle of the CLK2 ECL-input signal range.

2.3.1 RCLK, SCLK, VCLK

The Viewpoint VIP provides user-programmable reference clock (RCLK), shift clock (SCLK), and video (VCLK) clock outputs that can be set as divisions of the dot clock. RCLK is a continuously running reference clock and is not disabled during blank. RCLK can be selected as divisions of 1, 2, 4, 8, 16, 32 or 64 of the dot clock (see Table 2–5). It is provided as a clock reference and is typically connected back to the LCLK input to latch pixel-port data. Since pixel-port data is latched on the rising edge of LCLK, the RCLK frequency must be set as a function of the desired multiplexing ratio (that depends on the pixel bus width and number of bit planes). See Section 2.4 and Table 2–6.

SCLK is the same as RCLK but disabled during the blank active period. SCLK is designed to be used as the shift clock to interface directly with the VRAM. If SCLK is not used, the output can be switched off and held low to protect against VRAM lockup due to invalid SCLK frequencies. The detailed SCLK control timing is discussed in Section 2.3.2 and illustrated in Figures 2–2 through 2–5.

VCLK is designed to be used as the timing reference by the graphics processor or other custom-designed control logic to generate the graphics system control signals ($\overline{\text{SYSBL}}$, $\overline{\text{SYSHS}}$, and $\overline{\text{SYSVS}}$). VCLK can be selected as divisions of 1, 2, 4, 8, 16, 32, or 64 of the dot clock and can also be held at logic 1 (see Table 2–5). The default setup is VCLK held at logic 1 since it is not used in VGA pass-through mode. Since these control signals are sampled by VCLK, VCLK must be enabled for these to function properly (see Figures 2–2 through 2–5).

Even though RCLK/SCLK and VCLK can be selected independently, there is still a relationship between the two as discussed below. Many system considerations have been carefully covered in their design, leaving maximum freedom to the user.

Internally, RCLK, SCLK, and VCLK are generated from a common clock counter that is counted at the rising edge of the DOTCLK. Therefore, when VCLK is enabled, it is naturally in phase with RCLK and SCLK as shown in Figure 2–1.

The internal clock counter is initialized any time the output-clock-selection register is written with 3F (hex). This provides a simple mechanism to synchronize multiple palettes or system devices by providing a known phase relationship for the various system clocks. It is left up to the user to provide some means of disabling the dot clock input to the part while this reset is occurring if multiple parts are to be synchronized.

The reset default divide ratio for RCLK is 64:1 with SCLK held at logic 0 and VCLK held at logic 1.

When choosing certain video timing parameters, exercise caution if the selected RCLK frequency is less than the selected VCLK frequency. Refer to Appendix B for a more detailed discussion.

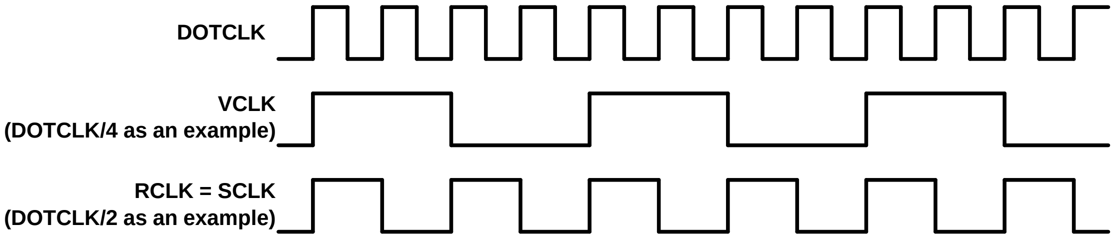


Figure 2–1. DOTCLK/VCLK/RCLK/SCLK Relationship

The input-clock-selection register is used to select the desired input clock source. Table 2–4 details how to program the various options.

Table 2–4. Input-Clock-Selection Register

INPUT-CLOCK-SELECT REGISTER (HEX) (see Note 1)	FUNCTION (see Note 2)
00	Select CLK0 as clock source[†]
01	Select CLK1 as clock source
02	Select CLK2 as TTL clock source
03	Select $\overline{\text{CLK2}}$ as TTL clock source
04	Select CLK2 and $\overline{\text{CLK2}}$ as ECL clock source
10	Select CLK0 as doubled clock source
11	Select CLK1 as doubled clock source
12	Select CLK2 as TTL doubled clock source
13	Select CLK2 as TTL doubled clock source
14	Select CLK2/$\overline{\text{CLK2}}$ as ECL doubled clock source

[†] CLK0 is chosen at RESET as required for VGA pass-through.

NOTES: 1. Register bits 3 and 7 are don’t-care bits.
2. Register bits 5 and 6 are reserved.
3. When the clocks are selected from one input clock source to another, a minimum of 30 ns is needed before the new clocks are stabilized and running.

The output-clock-selection register is used to program the desired divided-down frequencies for the reference/shift and video clocks.

Table 2–5. Output-Clock-Selection Register Format

OUTPUT-CLOCK-SELECTION-REGISTER BITS (see Note 4)						FUNCTION (see Notes 4, 5, 6, and 7)
5	**4**	**3**	**2**	**1**	**0**	
0	0	0	x	x	x	VCLK/1 output ratio
0	0	1	x	x	x	VCLK/2 output ratio
0	1	0	x	x	x	VCLK/4 output ration
0	1	1	x	x	x	VCLK/8 output ratio
1	0	0	x	x	x	VCLK/16 output ratio
1	0	1	x	x	x	VCLK/32 output ratio
1	1	0	x	x	x	VCLK/64 output ratio
1	1	1	x	x	x	VCLK output held at logic 1[‡]
x	x	x	0	0	0	RCLK/1 output ratio (see Notes 4 and 7)
x	x	x	0	0	1	RCLK/2 output ratio (see Notes 4 and 7)
x	x	x	0	1	0	RCLK/4 output ratio (see Notes 4 and 7)
x	x	x	0	1	1	RCLK/8 output ratio (see Notes 4 and 7)
x	x	x	1	0	0	RCLK/16 output ratio (see Notes 4 and 7)
x	x	x	1	0	1	RCLK/32 output ratio (see Notes 4 and 7)
x	x	x	1	1	0	RCLK/64 output ratio (see Notes 4 and 7)

[‡] These lines indicate the RESET conditions as required for VGA pass-through mode.

NOTES: 4. Register bit 6 enables (logic 1) and disables (default – logic 0) the SCLK output buffer. Register bit 7 is a don’t-care bit.
5. When the clocks are selected from one mode to the other, a minimum of 30 ns is needed before the new clocks are stabilized and running.
6. When the output-clock-selection register is written with 3F (hex), the clock counter is reset, RCLK = SCLK = logic 0, and VCLK = logic 1.
7. SCLK is the same as RCLK except that it is disabled during blank. When the RCLK divide ratio is chosen, this sets the SCLK ratio as well.

Table 2–5. Output-Clock-Selection Register Format (Continued)

OUTPUT-CLOCK-SELECTION REGISTER BITS (see Note 4)							FUNCTION (see Notes 5, 6, and 7)
6	5	4	3	2	1	0	
0	x	x	x	1	1	0	RCLK/64, SCLK output held at logic 0†
0	x	x	x	1	1	1	RCLK, SCLK outputs held at logic 0
x	1	1	1	1	1	1	Clock counter reset (5)

† These lines indicate the RESET conditions as required for VGA pass-through.

NOTES: 4 Register bit 6 enables (logic 1) and disables (default – logic 0) the SCLK output buffer. Register bit 7 is a don't-care bit.

5 When the clocks are selected from one mode to the other, a minimum of 30 ns is needed before the new clocks are stabilized and running.

6 When the output-clock-selection register is written with 3F (hex), the clock counter is reset, RCLK = SCLK = logic 0, and VCLK = logic 1.

7 SCLK is the same as RCLK except that it is disabled during blank. When the RCLK divide ratio is chosen, this sets the SCLK ratio as well.

2.3.2 Frame-Buffer Clocking: Self or Externally Clocked

The Viewpoint VIP has two pixel-data latching modes, allowing for flexibility in the frame-buffer interface timing. For the pixel port (P0–P63), data is always latched on the rising edge of LCLK. If auxiliary-control register (ACR) bit 3 is set to logic 1 (default), the internal circuitry is configured for self-clocked mode. In this mode, the RCLK or SCLK output of the palette must be used as the timing reference to present data to the pixel port (P0–P63). In self-clocked mode, RCLK can be directly tied back to LCLK or LCLK can be a delayed version of RCLK within the timing requirements of the TVP3020. The self-clocked mode of frame-buffer latching is similar to the operation of the TLC3407X video interface palette devices.

The internal Viewpoint blank signal is generated from either $\overline{\text{VGABL}}$ or $\overline{\text{SYSBL}}$, depending on whether the VGA port is enabled (multiplex-control register 2 (MCR2) bit 7 = logic 1) or disabled (MCR2 bit 7 = logic 0). The rising edge of CLK0 is used to latch $\overline{\text{VGABL}}$ when the VGA port is enabled. The falling edge of VCLK is used to sample and latch the $\overline{\text{SYSBL}}$ input when the VGA port is disabled. When the internal blank becomes active, SCLK is disabled as soon as possible. For example, if SCLK is high when the sampled $\overline{\text{SYSBL}}$ goes low, SCLK is allowed to complete the clock cycle and return to the low state. SCLK then is held low until the sampled $\overline{\text{SYSBL}}$ signal goes back high. At this time, SCLK is enabled to clock the first pixel data valid from VRAM. The Viewpoint video blanking circuitry is designed with sufficient pipeline delay to allow the internal sampled $\overline{\text{SYSBL}}$ and $\overline{\text{VGABL}}$ signals to align with the pipelined RGB data to the video DACs. The logic described above works in situations where the SCLK period is shorter than, equal to, or longer than the VCLK period.

When in the self-clocked mode, the SCLK control timing is designed to interface directly with the external VRAM. The shift register in the system VRAM is supposed to be updated during the blank active period. When the $\overline{\text{SYSBL}}$ input is sampled high by the falling edge of VCLK, the VRAM shift clock (SCLK) is restarted to clock the VRAM and enable the first group of pixel data to appear on the pixel bus as well as at the Viewpoint pixel input port. The second SCLK causes the VRAM shift register to shift out the second group of data. At the same time, LCLK latches the first group of pixel data into the Viewpoint palette (refer to Figure 2–2 for a detailed timing diagram).

The RCLK/SCLK phase relationship is designed such that timing specifications are satisfied for the case where SCLK is driving a typical 2-MB VRAM load and RCLK is connected to LCLK. If an external buffer is required on SCLK so that it can drive a larger load, a similar buffer can be placed on RCLK to match the signal delay before connecting to LCLK. However, the delay from LCLK to RCLK cannot exceed one RCLK period –5 ns. Please refer to the timing parameter specifications for more details.

When the VRAM split-shift-register operation is performed (see Figure 2–3), the SCLK timing is adjusted to work with the SFLAG input. Basically, the split-shift-register operation inserts an SCLK during the blank period. This causes the first group of pixel data to appear at the pixel port during blank and allows the first group of data to be displayed as soon as the palette comes out of blank. Figures 2–3 and 2–5 show the case when the SSRT (split shift-register transfer) function is enabled. When a rising edge occurs on the SFLAG input, one SCLK with a minimum of 15-ns pulse duration is generated after the specified delay. Since this is designed to meet VRAM timing requirements, the SSRT-generated SCLK replaces the first SCLK in the regular shift register transfer case as described above. Refer to Section 2.15 for a detailed explanation of the SSRT function.

Externally clocked timing can be chosen for the pixel bus (P0 – P63) by setting auxiliary control register bit 3 to a logic 0. In externally clocked mode, the RCLK or SCLK output of the palette is not used as the timing reference to present data to the pixel bus. Instead, pixel data is presented to the palette with a synchronous clock and all palette timing is referenced to this clock. In this mode, the external clock should be connected to LCLK and the selected clock input. (If the VGA port is enabled, the CLK0 input is selected independent of the input clock selection register.)

The externally clocked frame-buffer interface mode is intended for applications where windowed or pixel-by-pixel switching between the VGA port and the pixel port is desired in non-VRAM-based graphics systems. In such applications, the VGA port is enabled (multiplex-control register bit 7 set to logic 1) and the appropriate direct color mode is set in the multiplex-control register. The auxiliary window, port select, and/or color-key switching functions are then configured and enabled to perform the desired switching. By setting the frame-buffer interface to the externally clocked mode, the pixel port and VGA port timing and pipeline delay are made the same. Also, since the VGA port is enabled, all video control signal timing is referenced to CLK0, utilizing the $\overline{\text{VGABL}}$, VGAHS, and VGAVS inputs.

The externally clocked frame-buffer interface timing can also be used in non-VGA switching applications, utilizing only the pixel port or only the VGA port. In either case, it is recommended that VGA video control signals be used (i.e., $\overline{\text{VGABL}}$, VGAHS, VGAVS). In this way, all pixel data and video control signals are referenced to CLK0, and video blank and sync are aligned with pixel data.

> **NOTE:** If the pixel port is used in externally clocked mode (ACR3 = 0), RCLK must be set to DOT/1 in the output-clock-selection register and a 1:1 multiplexing mode must be selected in the multiplexer control registers (see Table 2–6). The external clock should be connected to the LCLK input as well as the selected clock input. If the VGA port is also enabled (MCRB7 = 1), CLK0 is selected as the input clock independent of the input-clock-selection register setting.
>
> VGA switching can only be performed using a 1:1 multiplexing mode.
>
> Overlay switching can only be performed using a 1:1 multiplexing mode if the pixel port is set for externally clocked mode. If the pixel port is self-clocked, any of the multiplex ratios available in Table 2–6 may be used.
>
> If VGA switching is to be performed using externally clocked mode (ACR3 = 0), the full VGA port frequency of 85 MHz may be utilized provided that the VGA port and the pixel port are both synchronized to the CLK0 input clock.
>
> If VGA switching is to be performed using self-clocked mode (ACR3 = 1), the maximum pixel rate cannot exceed 50 MHz. This is because of internal delay from the CLK0 input to the RCLK output.

VGA data pipeline delay is adjusted within the Viewpoint VIP depending on whether self- or externally-clocked frame-buffer interface timing is used (see Section 2.3.2). If the Viewpoint palette is programmed for self-clocked timing, three additional dot clock pipeline delays are inserted into the internal VGA data path and into the internal blanking signal. The additional pipeline delay accounts for the difference between $\overline{\text{VGABL}}$ or $\overline{\text{SYSBL}}$ and the pixel data inputs (P0 – P63) when used in the self- and externally-clocked modes. This is so the VGA and pixel-port data remain synchronous in time when doing auxiliary window, port select, or color-keyed switching (see Section 2.6). If externally clocked timing is used, the VGA port and the pixel port are already synchronous since both data and blanking are presented to the palette during the same CLK0 clock cycle.

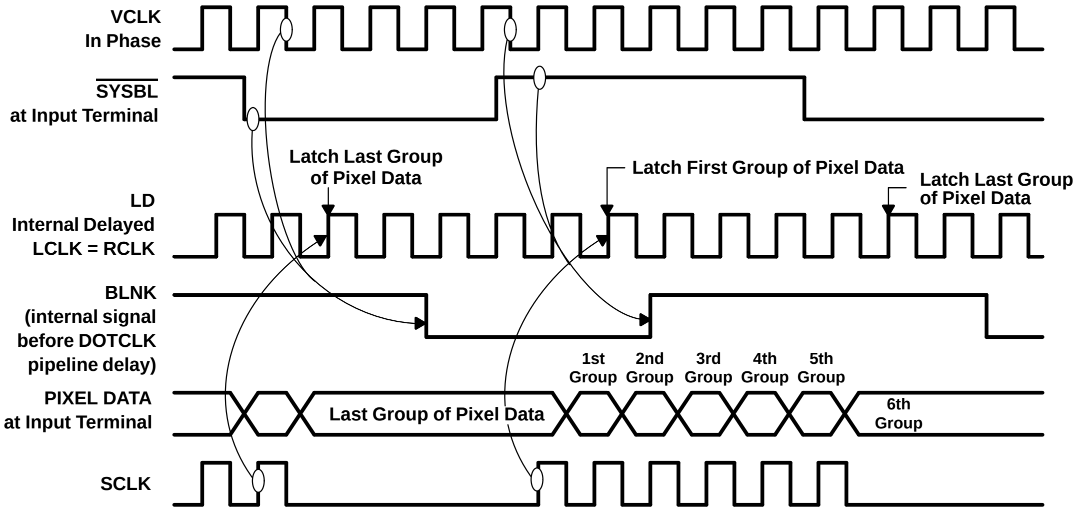


Figure 2–2. SCLK/VCLK Control Timing
(SSRT Disabled, RCLK/SCLK Frequency = VCLK Frequency)

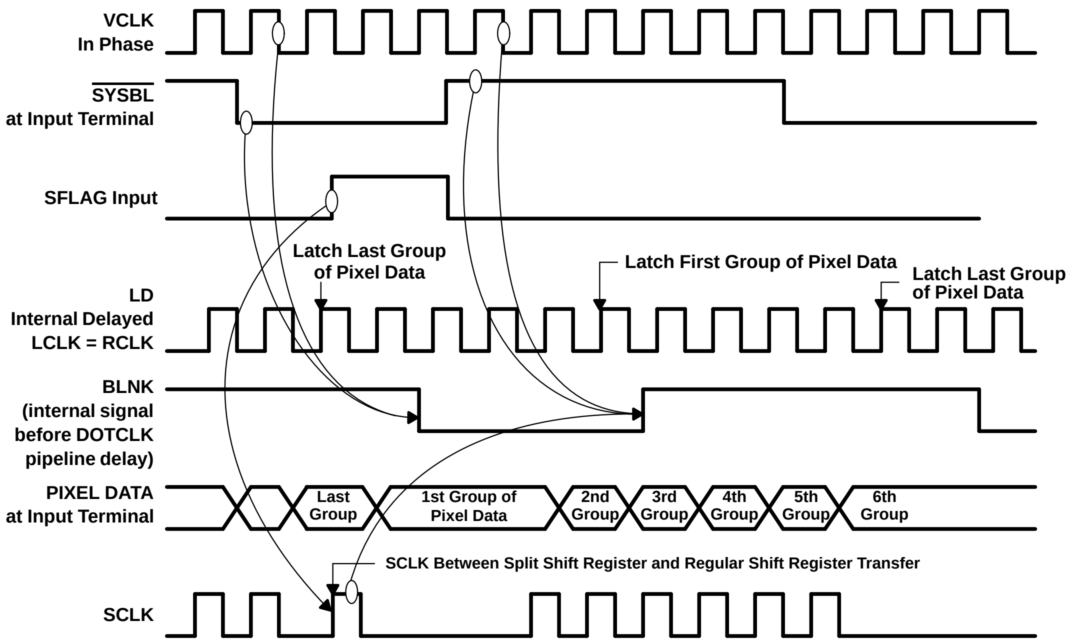


Figure 2–3. SCLK/VCLK Control Timing
(SSRT Enabled, RCLK/SCLK Frequency = VCLK Frequency)

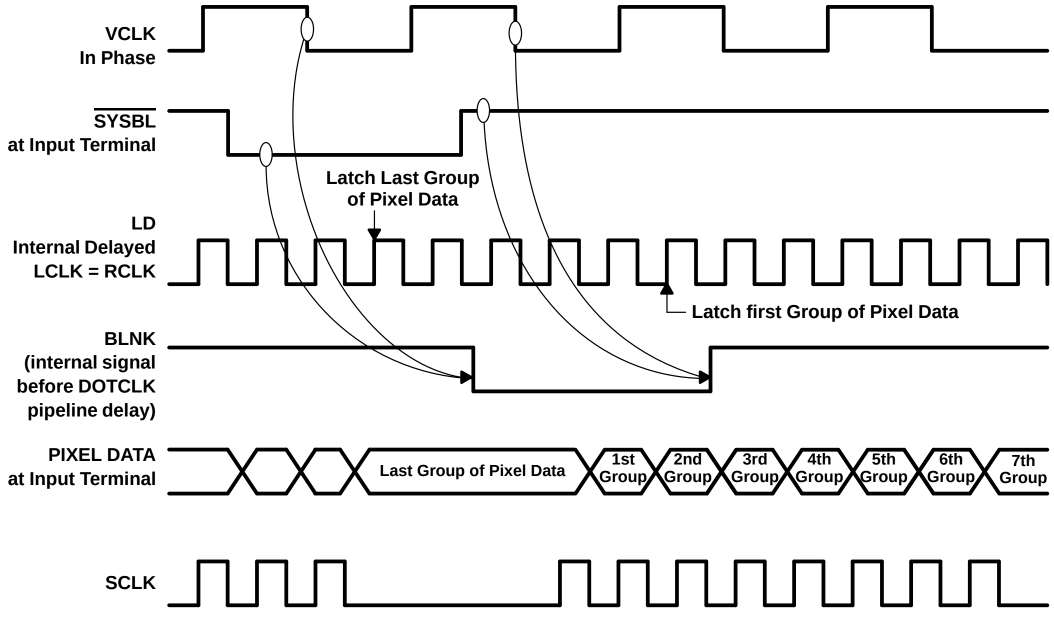


Figure 2–4. SCLK/VCLK Control Timing
(SSRT Disabled, RCLK/SCLK Frequency = 4 x VCLK Frequency)

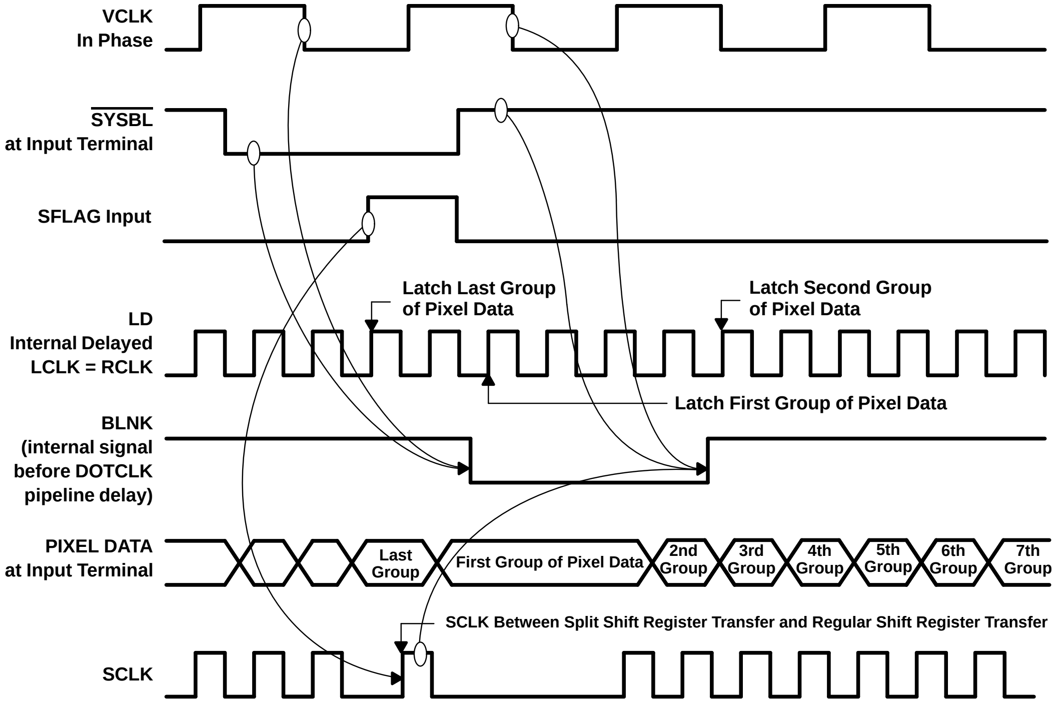


Figure 2–5. SCLK/VCLK Control Timing (SSRT Enabled, RCLK/SCLK Frequency = 4 x VCLK Frequency)

2.4 Multiplexing Scheme

The Viewpoint palette offers a highly versatile multiplexing scheme as illustrated in Tables 2–6 through 2–11. The multiplexing scheme allows the pixel bus to be programmed to 1, 2, 4, 8, 12, 16, 24, or 32 bits/pixel with pixel bus widths ranging from 1 bit to 64 bits. The use of on-chip multiplexing allows graphics systems to be designed that can support multiple pixel depths and resolutions with no hardware modification. It also allows the system to be configured to the amount of RAM available. For example, if only 256K bytes of memory are available, an 800-by-600 mode with 4 bit planes (four bits per pixel) could be implemented using an 8-bit-wide pixel bus. If at a later date another 256K bytes are added to another 8 bits of the pixel bus, the user has the option of using 8 bit planes at the same resolution or 4 bit planes at a 1024 by 768 resolution. When a further 512K bytes are added to the remaining 16 bits of the pixel bus, the user has the option of 8 bit planes at 1024 by 768 or 4 bit planes at 1280 by 1024. The Viewpoint palette can also be configured for direct-color or true-color operation. All of the above could be achieved without any board-level hardware modification and without any increase in the speed of the pixel bus.

Multiplexing of the pixel bus is controlled by and programmed through multiplex-control registers 1 and 2. Table 2–6 details the multiplex-control register settings for each mode of operation (see also Sections 2.4.2 through 2.4.6).

2.4.1 Little-Endian and Big-Endian Data Format

The Viewpoint pixel bus supports both little- and big-endian data formats for all pseudo-color, direct-color, and true-color modes of operation. The data-format select is controlled by general-control register bit 3 (see Section 2.16.1). When general-control register (GCR) bit 3 is set to 0 (default), then the format is set to little endian. When GCR bit 3 is set to 1, then the format is set to big endian.

In a big-endian design, the external VRAM data bus bits must be connected in reverse order to the Viewpoint pixel bus; i.e., D63 connected to P0, D0 connected to P63, etc. This ensures that the least significant channel always provides the first pixel to be displayed in the pseudo-color or true-color multiplexing modes. The difference between little- and big-endian data formats and how they affect the pixel bus operation is discussed in detail in Appendix C.

2.4.2 VGA Pass-Through Mode

The VGA pass-through mode is used to emulate the VGA modes of most personal computers. The advantage of this mode is that it can take data presented on the feature connector of most VGA-compatible PC systems into the device on a separate bus, thus requiring no external multiplexing. This feature is particularly useful in systems where the existing graphics circuitry is on the motherboard. In this instance, it enables a drop-in graphics card to be implemented that maintains compatibility with all existing software. This is accomplished by using the on-board VGA circuitry but routing the emerging bit-plane data through the Viewpoint palette. VGA pass-through is the default mode at RESET. If VGA pass-through is desired at power up, an external resistor, capacitor, and diode network should be connected to $\overline{\text{RESET}}$ (see Figure A–1).

Since this mode is designed with the feature connector philosophy, all data latching and control timing is referenced to CLK0. When the VGA port is enabled (MCR2 bit 7 = 1), CLK0 is selected as the input clock source independent of the input-clock-selection register. The VGA port always operates as in the externally clocked mode of the pixel port (P0–P63); it receives the VGA data (VGA0 – VGA7) and the VGA blank ($\overline{\text{VGABL}}$), both of which are referenced to an external clock (CLK0). CLK0 also latches the $\overline{\text{VGABL}}$, VGAHS, and VGAVS video control signals when in the VGA pass-through mode. External signals on LCLK have no effect on the VGA port since LCLK only latches data on the pixel port (pins P0–P63).

VGA data pipeline delay is adjusted within the Viewpoint palette depending on whether self- or externally-clocked frame-buffer interface timing is used (see Section 2.3.2). If TVP3020 is programmed for self-clocked timing, additional dot clock pipeline delay is inserted into the internal VGA data path. The purpose is so that the VGA and pixel port data can remain synchronous when doing auxiliary window, port select, or color-keyed switching (see Section 2.6). The additional VGA pipeline delay accounts for the dot clock to RCLK pipeline delay within the palette.

2.4.3 Pseudo-Color Mode

In pseudo-color mode (sometimes called color indexing), the pixel-bus inputs are used to address the palette-RAM LUT (color look up table). The data in each RAM location is comprised of 24 bits (8 bits for each of the red, green, and blue color DACs). The pseudo-color mode is further grouped into 4 submodes, depending on the data bits per pixel. In each submode, a pixel bus width of 4, 8, 16, 32 or 64 bits may be used. Data should always be presented on the least significant bits of the pixel bus; i.e., when 16 bits are used, the pixel data must be presented on P15–P0, 8 bits on P7–P0, and 4 bits on P3–P0. See Tables 2–6 through 2–11 for more details.

Submode 1 uses a single bit plane to address the color palette. The pixel port bit is fed into bit 0 of the palette address, with the 7 high-order address bits defined by the palette-page register (see Section 2.2.3). This mode has uses in high-resolution monochrome applications such as desktop publishing. This mode allows the maximum amount of multiplexing with 64:1 ratio, thus giving a pixel bus rate of only 2 MHz at a screen resolution of 1280 by 1024. Although only a single bit is used, alteration of the palette-page register at the line frequency allows 256 different colors to be displayed per screen with two colors per line.

Submode 2 uses two bit planes to address the color palette. The two bits are fed into the low-order address bits of the palette with the six high-order address bits being defined by the palette-page register (see Section 2.2.3). This mode allows a maximum multiplex ratio of 32:1 on the pixel bus and is essentially a four-color alternative to submode 1.

Submode 3 uses four bit planes to address the color palette. The four bits are fed into the low-order address bits of the palette with the four high-order address bits being defined by the palette-page register (see Section 2.2.3). This mode provides 16 pages of 16 colors and can be used at multiplex ratios of /1 to /16.

Submode 4 uses eight bit planes to address the color palette. Since all eight bits of palette address are specified from the pixel port, the page register is not used. This mode allows dot clock-to-LCLK ratios of 1:1 (8-bit bus), 2:1 (16-bit bus), 4:1 (32-bit bus) or 8:1 (64-bit bus). Therefore, in a 64-bit configuration, a 1024 by 768 pixel screen can be implemented with an external data rate of only 8 MHz.

> **NOTE:** If externally clocked-frame buffer timing is used (ACR3 = logic 0, see Section 2.3.2), only multiplex ratios of 1:1 can be used. See Table 2–6.
>
> The auxiliary window, port select, and color-key switching functions must be disabled and set for palette graphics when in the pseudo-color mode. This is the default condition at RESET. See Section 2.6

2.4.4 Direct-Color Mode

In direct-color mode, 24, 16, 15, or 12 bits of data can be transferred directly to the RGB DACs but with the same amount of pipeline delay as the overlay data and the control signals (blank and SYNCs). Depending on which direct-color mode is selected, overlay is provided by utilizing the remaining bits of the pixel bus to address the palette RAM. This results in a 24-bit RAM output that is then used as overlay information to the DACs. The overlay capability is designed to work with the auxiliary window, port select, and color-key switching functions to provide overlay in specific windows or on a pixel-by-pixel basis on the direct-color display as discussed in Section 2.6. See Tables 2–6, 2–8, and 2–9 for more details on selecting the direct-color modes.

The default condition after reset is for the auxiliary window and port select functions to be disabled (ACR1 = ACR2 = logic 0). The color-key comparisons, which are controlled by the color-key control (CKC) register bits 0–3, are also disabled (CKC0 = CKC1 = CKC2 = CKC3 = logic 0). Also, since multiplex-control register 2 bit 7 = logic 1 and ACR0 = CKC4 = logic 1 at RESET, the default is for VGA pass through. This is because multiplex-control register 2 bit 7 enables the VGA port and the switching functions (SWITCH = COLOR-KEY = 1, see Section 2.6) are disabled and set for palette graphics (as opposed to direct color-palette bypass).

Submode 1 is the 24-bit direct-color mode that uses eight bits to represent each color and eight bits for overlay. The 64-bit-wide pixel bus of the Viewpoint palette allows multiplex ratios of 1:1 or 2:1. In this mode, there are basically two different configurations: either the 32-bit data is grouped as overlay, red, green, blue, or blue, green, red, overlay.

Submode 2 is the XGA-compatible (5–6–5) 16-bit-color mode supporting five bits of red, six bits of green, and five bits of blue data. Viewpoint supports multiplex ratios for this mode of 1:1, 2:1, and 4:1. With 4:1 multiplexing, the TVP3020 can display 1280 x 1024 direct-color using 45-MHz VRAM without any glue logic. Overlay is not available in this mode.

Submode 3 is the TARGA-compatible (5–5–5) mode that uses 15 bits for color and 1 bit for overlay. It allows 5 bits for each of red, green, and blue data. The TVP3020 supports 1:1, 2:1, and 4:1 multiplexing ratios in this mode.

Submode 4 is (6–6–4) configuration. It provides six bits of red, six bits of green, and four bits of blue. The TVP3020 also supports 1:1, 2:1, and 4:1 multiplexing in this mode. Overlay is not available in this mode.

Submode 5 is (4–4–4–4) configuration. It provides 12 bits of direct color and 4 bits of overlay. It allows four bits for each of red, green, and blue data. The TVP3020 supports 1:1, 2:1, and 4:1 multiplexing ratios in this mode.

See NOTES in the following section.

2.4.5 True-Color Mode

In true-color mode, the palette RAM is partitioned into three independent 256-word × 8-bit memory blocks that can be individually addressed by each color field of the true-color data. The independent memory blocks provide data for a single DAC output. With this architecture, gamma correction for each color is possible. Since the palette is used in true-color mode, there is no memory space to be used for the overlay function. All of the true-color submodes are the same as direct color except that overlay is not available. See Tables 2–6 through 2–11 for more details on mode selection. See notes below.

> **NOTES:** Since less than 8 bits are defined for each color in the various 12- or 16-bit direct- or true-color modes, the data bits for the individual colors are internally shifted to the MSB locations and the remaining LSB locations for each color are set to logic 0 before 8-bit data is sent to the DACs.
>
> Since the overlay information goes through the pseudo-color data path, it is subject to read masking and the palette-page register. This is especially important for those direct-color modes that have less than eight bits of overlay information. The overlay information in these modes justifies to the LSB bit positions, and the remaining MSB positions are filled with the corresponding palette-page data before addressing the palette RAM.
>
> In order to display true color (gamma corrected through the palette), either the auxiliary windowing or the color-key switching function must be set for palette graphics. For direct color, both functions must be set for direct color.
>
> In order to use the overlay capability of the direct-color modes, the color-key switching or port-select function must be configured and enabled. Overlay port data in a window is also available by enabling the auxiliary window function. If either the auxiliary windowing or the color-key switching functions point to palette graphics, palette graphics are always displayed (not direct color).
>
> When in the 24-bit direct-color or true-color modes, the data input works only in the 8-bit mode. In other words, if only six bits are used, the two LSB inputs for each color need to be tied to GND. However, the palette, which is used by the overlay input, is still governed by $8/\overline{6}$, and the output multiplexer selects 8 bits or 6 bits of data accordingly. The $8/\overline{6}$ is also valid in the other 16-bit modes.
>
> The definitions of direct color (palette bypass) and true color are consistent with the IBM XGA terminology.

2.4.6 Multiplex-Control Registers

The pixel port multiplexer is controlled by two 8-bit registers in the indirect register map (see Section 2.1). The various multiplexing modes can be selected according to the following table.

Table 2–6. Multiplex Mode and Bus-Width Selection

MODE	SUB-MODE	MULTIPLEX-CONTROL REGISTER 1 (HEX)	MULTIPLEX-CONTROL REGISTER 2 (HEX)	DATA BITS PER PIXEL (see Note 8)	PIXEL BUS WIDTH	MULTI-PLEX RATIO (see Note 9)	OVERLAY BITS PER PIXEL	TABLE REFERENCE (see Note 10)
VGA		80	98	8	8	1	NA	v1
Pseudo-Color	1	80	00	1	4	4	NA	s1
		80	01	1	8	8	NA	s2
		80	02	1	16	16	NA	s3
		80	03	1	32	32	NA	s4
		80	04	1	64	64	NA	s5
	2	80	08	2	4	2	NA	s6
		80	09	2	8	4	NA	s7
		80	0A	2	16	8	NA	s8
		80	0B	2	32	16	NA	s9
		80	0C	2	64	32	NA	s10
	3	80	10	4	4	1	NA	s11
		80	11	4	8	2	NA	s12
		80	12	4	16	4	NA	s13
		80	13	4	32	8	NA	s14
		80	14	4	64	16	NA	s15
	4	80	19	8	8	1	NA	s16
		80	1A	8	16	2	NA	s17
		80	1B	8	32	4	NA	s18
		80	1C	8	64	8	NA	s19

NOTES: 8. Data bits per pixel is the number of bits of pixel port information used as color data for each displayed pixel, often referred to as the number of bit planes.

9. Multiplex ratio indicates the number of pixels per bus load or the number of pixels associated with each LCLK (load clock) pulse. For example, with a 32-bit pixel bus width and 8 bit planes, each bus load consists of four pixels. In a typical implementation, the LCLK signal is either connected to or derived from RCLK. Therefore, the RCLK divide ratio must be chosen as a function of the multiplex mode selected. The RCLK divide ratio is not automatically set by mode selection but must be programmed in the output clock selection register by the user.

10. This column is a reference to Tables 2–7 through 2–11, where the actual manipulation of pixel information and pixel latching sequences are illustrated for each of the multiplexing modes.

11. It is recommended that all unused input terminals be connected to ground to conserve power.

12. Multiplex-control register 2 bit 7 enables (logic 1) and disables (logic 0) the VGA port. If auxiliary-window or port-select switching is to be done involving the VGA port, this bit needs to be set to a logic 1 as well as programming for the correct direct-color mode. For example, if auxiliary windowing is to be done with direct-color submode 1 (32-bit pixel bus) and VGA, instead of programming 1B (hex), multiplex-control register 2 should be programmed to 9B (hex). If only VGA pass-through is desired, the values should be programmed for VGA mode as indicated in Table 2–6. If only VGA pass-through is desired, the values should be programmed as indicated in Table 2–6 for VGA mode.

Table 2–6. Multiplex Mode and Bus-Width Selection (Continued)

MODE	SUB-MODE	MULTIPLEX-CONTROL REGISTER 1 (HEX)	MULTIPLEX-CONTROL REGISTER 2 (HEX)	DATA BITS PER PIXEL (see Note 8)	PIXEL BUS WIDTH	MULTIPLEX RATIO (see Note 9)	OVERLAY BITS PER PIXEL	TABLE REFERENCE (see Note 10)
Direct	1 24-bit	06	1B	24	32	1	8	d1
		06	1C	24	64	2	8	d2
		07	1B	24	32	1	8	d3
		07	1C	24	64	2	8	d4
	2 (5–6–5) XGA	05	02	16	16	1	NA	d5
		05	03	16	32	2	NA	d6
		05	04	16	64	4	NA	d7
	3 (5–5–5) TARGA	04	02	15	16	1	1	d8
		04	03	15	32	2	1	d9
		04	04	15	64	4	1	d10
	4 16-bit (6–6–4)	03	02	16	16	1	NA	d11
		03	03	16	32	2	NA	d12
		03	04	16	64	4	NA	d13
	5 12-bit (4–4–4)	01	12	12	16	1	4	d14
		01	13	12	32	2	4	d15
		01	14	12	64	4	4	d16

NOTES:

9. Data bits per pixel is the number of bits of pixel port information used as color data for each displayed pixel, often referred to as the number of bit planes.
10. Multiplex ratio indicates the number of pixels per bus load or the number of pixels associated with each LCLK (load clock) pulse. For example, with a 32-bit pixel bus width and 8 bit planes, each bus load consists of four pixels. In a typical implementation, the LCLK signal is either connected to or derived from RCLK. Therefore, the RCLK divide ratio must be chosen as a function of the multiplex mode selected. The RCLK divide ratio is not automatically set by mode selection but must be programmed in the output clock selection register by the user.
11. This column is a reference to Tables 2–8 through 2–11, where the actual manipulation of pixel information and pixel latching sequences are illustrated for each of the multiplexing modes.
12. It is recommended that all unused input terminals be connected to ground to conserve power.
13. Multiplex-control register 2 bit 7 enables (logic 1) and disables (logic 0) the VGA port. If auxiliary-window or port-select switching is to be done involving the VGA port, this bit needs to be set to a logic 1 as well as programming for the correct direct-color mode. For example, if auxiliary windowing is to be done with direct-color submode 1 (32-bit pixel bus) and VGA, instead of programming 1B (hex), multiplex-control register 2 should be programmed to 9B (hex). If only VGA pass-through is desired, the values should be programmed for VGA mode as indicated in Table 2–6. If only VGA pass-through is desired, the values should be programmed as indicated in Table 2–6 for VGA mode.

Table 2–6. Multiplex Mode and Bus-Width Selection (Continued)

MODE	SUB-MODE	MULTIPLEX-CONTROL REGISTER 1 (HEX)	MULTIPLEX-CONTROL REGISTER 2 (HEX)	DATA BITS PER PIXEL (see Note 8)	PIXEL BUS WIDTH	MULTI-PLEX RATIO (see Note 9)	OVERLAY BITS PER PIXEL	TABLE REFERENCE (see Note 10)
True-Color	1 24-bit	46	03	24	32	1	NA	t1
		46	04	24	64	2	NA	t2
		47	03	24	32	1	NA	t3
		47	04	24	64	2	NA	t4
	2 (5–6–5) XGA	45	02	16	16	1	NA	t5
		45	03	16	32	2	NA	t6
		45	04	16	64	4	NA	t7
	3 (5–5–5) TARGA	44	02	15	16	1	NA	t8
		44	03	15	32	2	NA	t9
		44	04	15	64	4	NA	t10
	4 16-bit (6–6–4)	43	02	16	16	1	NA	t11
		43	03	16	32	2	NA	t12
		43	04	16	64	4	NA	t13
	5 12-bit (4–4–4)	41	02	12	16	1	NA	t14
		41	03	12	32	2	NA	t15
		41	04	12	64	4	NA	t16

NOTES: 9. Data bits per pixel is the number of bits of pixel port information used as color data for each displayed pixel, often referred to as the number of bit planes.

10. Multiplex ratio indicates the number of pixels per bus load or the number of pixels associated with each LCLK (load clock) pulse. For example, with a 32-bit pixel bus width and 8 bit planes, each bus load consists of four pixels. In a typical implementation, the LCLK signal is either connected to or derived from RCLK. Therefore, the RCLK divide ratio must be chosen as a function of the multiplex mode selected. The RCLK divide ratio is not automatically set by mode selection but must be programmed in the output clock selection register by the user.

11. This column is a reference to Tables 2–8 through 2–11, where the actual manipulation of pixel information and pixel latching sequences are illustrated for each of the multiplexing modes.

12. It is recommended that all unused input terminals be connected to ground to conserve power.

13. Multiplex-control register 2 bit 7 enables (logic 1) and disables (logic 0) the VGA port. If auxiliary-window or port-select switching is to be done involving the VGA port, this bit needs to be set to a logic 1 as well as programming for the correct direct-color mode. For example, if auxiliary windowing is to be done with direct-color submode 1 (32-bit pixel bus) and VGA, instead of programming 1B (hex), multiplex-control register 2 should be programmed to 9B (hex). If only VGA pass-through is desired, the values should be programmed for VGA mode as indicated in Table 2–6. If only VGA pass-through is desired, the values should be programmed as indicated in Table 2–6 for VGA mode.

Table 2–7. Pseudo-Color Mode Pixel-Latching Sequence (see Notes 13 and 14)

v1	s1	s2	s3	s4	s5	s6	s7
VGA7–VGA0	P0	P0	P0	P0	P0	P1, P0	P1–P0
	P1	P1	P1	P1	P1	P3, P2	P3–P2
	P2	P2	P2	P2	P2		P5–P4
	P3	•	•	•	•		P7–P6
		•	•	•	•		
		P7	P15	P31	P63		

s8	s9	s10	s11	s12	s13	s14	s15
P1–P0	P1–P0	P1–P0	P3–P0	P3–P0	P3–P0	P3–P0	P3–P0
P3–P2	P3–P2	P3–P2		P7–P4	P7–P4	P7–P4	P7–P4
P5–P4	P5–P4	P5–P4			P11–P8	P11–P8	P11–P8
•	•	•			P15–P12	•	•
•	•	•				•	•
P15–P14	P31–P30	P63–P62				P31–P28	P63–P60

s16	s17	s18	s19
P7–P0	P7–P0	P7–P0	P7–P0
	P15–P8	P15–P8	P15–P8
		P23–P16	P23–P16
		P31–P24	•
			•
			P63–P56

NOTES: 13. The latching sequence is initiated by a rising edge on LCLK. For modes in which multiple groups of data are latched, the LCLK rising edge latches all the groups and the pixel clock shifts them out starting with the low-numbered group. For example, in pseudo-color submode 3 with a 16-bit pixel bus width, the rising edge of LCLK latches all the data groups shown above (s13) and the pixel clock shifts them out in the order P(3–0), P(7–4), P(11–8), and P(15–12). Note that each line in each subtable above represents one pixel.

14. When in the big-endian format (GCR3 =1), the pixel bus is externally swapped by the user (i.e., D63 connected to P0, D0 connected to P63). Since data is always shifted out from low-numbered groups to high-numbered groups, the external swapping of the pixel bus causes the groups to be shifted out in the correct order. However, for modes with more than one bit per pixel, the bits in each data group are reversed (LSB to MSB). This is internally corrected by the Viewpoint input multiplexer. The difference between big- and little-endian data formats and how they affect the pixel bus operation is discussed in more detail in Appendix C.

Table 2–8. Direct-Color Mode Pixel-Latching Sequence (Little-Endian) (see Note 15)

d1	d2
P31–P24(O), P23–P16(R), P15–P8(G), P7–P0(B) MSB LSB	P31–P24(O), P23–P16(R), P15–P8(G), P7–P0(B) P63–P56(O), P55–P48(R), P47–P40(G), P39–P32(B) MSB LSB
d3	**d4**
P31–P24(B), P23–P16(G), P15–P8(R), P7–P0(O) MSB LSB	P31–P24(B), P23–P16(G), P15–P8(R), P7–P0(O) P63–P56(B), P55–P48(G), P47–P40(R), P39–P32(O) MSB LSB
d5	**d6**
P15–P11(R), P10–P5(G), P4–P0(B) MSB LSB	P15–P11(R), P10–P5(G), P4–P0(B) P31–P27(R), P26–P21(G), P20–P16(B) MSB LSB
d7	**d8**
P15–P11(R), P10–P5(G), P4–P0(B) P31–P27(R), P26–P21(G), P20–P16(B) P47–P43(R), P42–P37(G), P36–P32(B) P63–P59(R), P58–P53(G), P52–P48(B) MSB LSB	P15(O), P14–P10(R), P9–P5(G), P4–P0(B) MSB LSB
d9	**d10**

P15(O), P14–P10(R), P9–P5(G), P4–P0(B)
P31(O), P30–P26(R), P25–P21(G), P20–P16(B)
MSB LSB
	P15(O), P14–P10(R), P9–P5(G), P4–P0(B)
	P31(O), P30–P26(R), P25–P21(G), P20–P16(B)
	P47(O), P46–P42(R), P41–P37(G), P36–P32(B)
	P63(O), P62–P58(R), P57–P53(G), P52–P48(B)
	MSB LSB

d11	**d12**
P15–P10(R), P9–P4(G), P3–P0(B) MSB LSB	P15–P10(R), P9–P4(G), P3–P0(B) P31–P26(R), P25–P20(G), P19–P16(B) MSB LSB
d13	**d14**
P15–P10(R), P9–P4(G), P3–P0(B) P31–P26(R), P25–P20(G), P19–P16(B) P47–P42(R), P41–P36(G), P35–P32(B) P63–P58(R), P57–P52(G), P12–P48(B) MSB LSB	P15–P12(R), P11–P8(G), P7–P4(B), P3–P0(O) MSB LSB
d15	**d16**
P15–P12(R), P11–P8(G), P7–P4(B), P3–P0(O) P31–P28(R), P27–P24(G), P23–P20(B), P19–P16(O) MSB LSB	P15–P12(R), P11–P8(G), P7–P4(B), P3–P0(O) P31–P28(R), P27–P24(G), P23–P20(B), P19–P16(O) P47–P44(R), P43–P40(G), P39–P36(B), P35–P32(O) P63–P60(R), P59–P56(G), P55–P52(B), P51–P48(O) MSB LSB

NOTE 15: The latching sequence is initiated by a rising edge on LCLK. For modes in which multiple pixel data groups are latched on one LCLK rising edge, the pixel clock shifts them out starting with the low-numbered pixel data group. Note that each line of each subtable above represents one pixel.

Table 2–9. Direct-Color Mode Pixel-Latching Sequence (Big-Endian) (see Notes 16 and 17)

d1	**d2**
P31 – P24(B), P23 – P16(G), P15 – P8(R), P7 – P0(O)	P31 – P24(B), P23 – P16(G), P15 – P8(R), P7 – P0(O) P63 – P56(B), P55 – P48(G), P47 – P40(R), P39 – P32(O)
MSB LSB	MSB LSB
d3	**d4**
P31 – P24(O), P23 – P16(R), P15 – P8(G), P7 – P0(B)	P31 – P24(O), P23 – P16(R), P15 – P8(G), P7 – P0(B) P63 – P56(O), P55 – P48(R), P47 – P40(G), P39 – P32(B)
MSB LSB	MSB LSB
d5	**d6**
P15 – P11(B), P10 – P5(G), P4 – P0(R)	P15 – P11(B), P10 – P5(G), P4 – P0(R) P31 – P27(B), P26 – P21(G), P20 – P16(R)
MSB LSB	MSB LSB
d7	**d8**
P15 – P11(B), P10 – P5(G), P4 – P0(R) P31 – P27(B), P26 – P21(G), P20 – P16(R) P47 – P43(B), P42 – P37(G), P36 – P32(R) P63 – P59(B), P58 – P53(G), P52 – P48(R)	P15 – P11(B),P10 – P6(G),P5 – P1(R),P0(O)
MSB LSB	MSB LSB
d9	**d10**
P15 – P11(B), P10 – P6(G), P5 – P1(R), P0(O) P31 – P27(B), P26 – P22(G), P21 – P17(R), P16(O)	P15 – 11(B), P10 – P6(G), P5 – P1(R), P0(O) P31 – P27(B), P26 – P22(G), P21 – P17(R), P16(O) P47 – P43(B), P42 – P38(G), P37 – P33(R), P32(O) P63 – P59(B), P58 – P54(G), P53 – P49(R), P48(O)
MSB LSB	MSB LSB
d11	**d12**
P15 – P12(B), P11 – P6(G), P5 – P0(R)	P15 – P12(B), P11 – P6(G), P5 – P0(R) P31 – P28(B), P27 – P22(G), P21 – P16(R)
MSB LSB	MSB LSB
d13	**d14**
P15 – P12(B), P11 – P6(G), P5 – P0(R) P31 – P28(B), P27 – P22(G), P21 – P16(R) P47 – P44(B), P43 – P38(G), P37 – P32(R) P63 – P60(B), P59 – P54(G), P53 – P48(R)	P15 – P12(O), P11 – P8(B), P7 – P4(G), P3 – P0(R)
MSB LSB	MSB LSB
d15	**d16**
P15 – P12(O), P11 – P8(B), P7 – P4(G), P3 – P0(R) P31 – P28(O), P27 – P24(B), P23 – P20(G), P19 – P16(R)	P15 – P12(O), P11 – P8(B), P7 – P4(G), P3 – P0(R) P31 – P28(O), P27 – P24(B), P23 – P20(G), P19 – P16(R) P47 – P44(O), P43 – P40(B), P39 – P36(G), P35 – P32(R) P63 – P60(O), P59 – P56(B), P55 – P52(G), P51 – P48(R)
MSB LSB	MSB LSB

NOTES: 16. The latching sequence is the same as in the little-endian example. Each line represents one pixel.

17. These subtables assume that the pixel bus is externally reverse-wired for big-endian mode operation (i.e., D63 connected to P0, D0 connected to P63) and that big-endian mode has been selected in the general-control register (GCR3=1). The Viewpoint palette internally corrects the bit ordering (LSB to MSB) for each pixel.

Table 2–10. True-Color Mode Pixel-Latching Sequence (Little-Endian) (see Note 18)

t1	**t2**
P23 – P16(R), P15 – P8(G), P7 – P0(B) MSB LSB	P23 – P16(R), P15 – P8(G), P7 – P0(B) P55 – P48(R), P47 – P40(G), P39 – P32(B) MSB LSB
t3	**t4**
P31 – P24(B), P23 – P16(G), P15 – P8(R) MSB LSB	P31 – P24(B), P23 – P16(G), P15 – P8(R) P63 – P56(B), P55 – P48(G), P47 – P40(R) MSB LSB
t5	**t6**
P15 – P11(R), P10 – P5(G), P4 – P0(B) MSB LSB	P15 – P11(R), P10 – P5(G), P4 – P0(B) P31 – P27(R), P26 – P21(G), P20 – P16(B) MSB LSB
t7	**t8**
P15 – P11(R), P10 – P5(G), P4 – P0(B) P31 – P27(R), P26 – P21(G), P20 – P16(B) P47 – P43(R), P42 – P37(G), P36 – P32(B) P63 – P59(R), P58 – P53(G), P52 – P48(B) MSB LSB	P14 – P10(R),P9 – P5(G),P4 – P0(B) MSB LSB
t9	**t10**
P14 – P10(R), P9 – P5(G), P4 – P0(B) P30 – P26(R), P25 – P21(G), P20 – P16(B) MSB LSB	P14 – P10(R), P9 – P5(G), P4 – P0(B) P30 – P26(R), P25 – P21(G), P20 – P16(B) P46 – P42(R), P41 – P37(G), P36 – P32(B) P62 – P58(R), P57 – P53(G), P52 – P48(B) MSB LSB
t11	**t12**
P15 – P10(R), P9 – P4(G), P3 – P0(B) MSB LSB	P15 – P10(R), P9 – P4(G), P3 – P0(B) P31 – P26(R), P25 – P20(G), P19 – P16(B) MSB LSB
t13	**t14**
P15 – P10(R), P9 – P4(G), P3 – P0(B) P31 – P26(R), P25 – P20(G), P19 – P16(B) P47 – P42(R), P41 – P36(G), P35 – P32(B) P63 – P58(R), P57 – P52(G), P51 – P48(B) MSB LSB	P15 – P12(R), P11 – P8(G), P7 – P4(B) MSB LSB
t15	**t16**
P15 – P12(R), P11 – P8(G), P7 – P4(B) P31 – P28(R), P27 – P24(G), P23 – P20(B) MSB LSB	P15 – P12(R), P11 – P8(G), P7 – P4(B) P31 – P28(R), P27 – P24(G), P23 – P20(B) P47 – P44(R), P43 – P40(G), P39 – P36(B) P63 – P60(R), P59 – P56(G), P55 – P52(B) MSB LSB

NOTE 18: The latching sequence is initiated by a rising edge on LCLK. For modes in which multiple pixel data groups are latched on one LCLK rising edge, the pixel clock shifts them out starting with the low-numbered pixel data group. Note that each line in each subtable above represents one pixel.

Table 2–11. True-Color Mode Pixel-Latching Sequence (Big-Endian) (see Note 18)

t1	**t2**
P31 – P24(B), P23 – P16(G), P15 – P8(R) MSB LSB	P31 – P24(B), P23 – P16(G), P15 – P8(R) P63 – P56(B), P55 – P48(G), P47 – P40(R) MSB LSB
t3	**t4**
P23 – P16(R), P15 – P8(G), P7 – P0(B) MSB LSB	P23 – P16(R), P15 – P8(G), P7 – P0(B) P55 – P48(R), P47 – P40(G), P39 – P32(B) MSB LSB
t5	**t6**
P15 – P11(B), P10 – P5(G), P4 – P0(R) MSB LSB	P15 – P11(B), P10 – P5(G), P4 – P0(R) P31 – P27(B), P26 – P21(G), P20 – P16(R) MSB LSB
t7	**t8**
P15 – P11(B), P10 – P5(G), P4 – P0(R) P31 – P27(B), P26 – P21(G), P20 – P16(R) P47 – P43(B), P42 – P37(G), P36 – P32(R) P63 – P59(B), P58 – P53(G), P52 – P48(R) MSB LSB	P15 – P11(B),P10 – P6(G),P5 – P1(R) MSB LSB
t9	**t10**
P15 – P11(B), P10 – P6(G), P5 – P1(R) P31 – P27(B), P26 – P22(G), P21 – P17(R) MSB LSB	P15 – P11(B), P10 – P6(G), P5 – P1(R) P31 – P27(B), P26 – P22(G), P21 – 17(R) P47 – P43(B), P42 – P38(G), P37 – P33(R) P63 – P59(B), P58 – P54(G), P53 – P49(R) MSB LSB
t11	**t12**
P15 – P12(B), P11 – P6(G), P5 – P0(R) MSB LSB	P15 – P12(B), P11 – P6(G), P5 – P0(R) P31 – P28(B), P27 – P22(G), P21 – P16(R) MSB LSB
t13	**t14**
P15 – P12(B), P11 – P6(G), P5 – P0(R) P31 – P28(B), P27 – P22(G), P21 – P16(R) P47 – P44(B), P43 – P38(G), P37 – P32(R) P63 – P60(B), P59 – P54(G), P53 – P48(R) MSB LSB	P11 – P8(B), P7 – P4(G), P3 – P0(R) MSB LSB
t15	**t16**
P11 – P8(B), P7 – P4(G), P3 – P0(R) P27 – P24(B), P23 – P20(G), P19 – P16(R) MSB LSB	P11 – P8(B), P7 – P4(G), P3 – P0(R) P27 – P24(B), P23 – P20(G), P19 – P16(R) P43 – P40(B), P39 – P36(G), P35 – P32(R) P59 – P56(B), P55 – P52(G), P51 – P48(R) MSB LSB

NOTES: 17. The latching sequence is the same as in the little-endian example. Each line represents one pixel.

18. These subtables assume that the pixel bus is externally reverse wired for big-endian mode operation (i.e., D63 connected to P0, D0 connected to P63) and that big-endian mode has been selected in the general-control register (GCR3 = 1). The Viewpoint palette internally corrects the bit ordering (LSB to MSB) for each pixel.

2.5 On-Chip Cursor

The Viewpoint palette has an on-chip two-color 64 x 64 pixel user-definable cursor. The cursor operation defaults to the XGA standard, but X-Windows compatibility is also available (see Section 2.5.2). In addition to the 64 × 64 sprite cursor, TVP3020 also supports a two-color crosshair cursor. The cursors only operate in noninterlaced applications.

The pattern for the 64 × 64 cursor is provided by the cursor RAM, which may be accessed by the MPU at any time. Cursor positioning is performed via the cursor-position (x,y) registers and the sprite-origin (x,y) registers (see register bit definitions in Sections 2.16.3 and 2.16.4). Positions x and y are defined in the palette increasing from left to right and from top to bottom, respectively, as seen on the display screen. The cursor position (x,y) is relative to the first pixel displayed. In other words, the very first pixel displayed is located at position (0,0), and the last pixel displayed for a 1024 × 768 system is located at position (1023, 767).

On-chip cursor control is performed by the cursor-control register in the indirect register map (06 hex). Bits 0 and 1 control the width of the crosshair (1, 3, 5, or 7 pixels). Bit 2 enables/disables the crosshair cursor, and bit 3 controls the crosshair-cursor color. Bit 4 specifies either XGA or X-window mode for the sprite cursor. Bit 5 controls the color at the intersection of the sprite and crosshair cursors, and bit 6 enables/disables the sprite cursor. See the cursor-control register bit definitions in Section 2.16.2 for more details.

2.5.1 Cursor RAM

The 64 × 64 × 2 cursor RAM is used to define the pixel pattern within the 64 × 64 pixel cursor window. It is not initialized and may be written to or read by the MPU at any time. The cursor-RAM address zero is at the top left corner of the RAM as shown in Figure 2–6.

The cursor RAM is written to by loading a number into the cursor RAM. Address registers 09 and 08 (hex) of the index register indicate the location of the first group of four cursor pixels to be updated (two bits per pixel implies four pixels per byte). Then the first four pixels are written to the cursor-RAM data register 0A (hex) of the index register. This stores the cursor pixel data in the cursor RAM and automatically increments the cursor-RAM address register. A second write to the cursor-RAM data register then loads the next four cursor pixels, and so on. See the register bit definitions in Sections 2.16.8 and 2.16.9.

To read from the cursor RAM, the address of the first cursor-RAM location to be read is loaded into the cursor-RAM address registers. Then a read is performed on the cursor-RAM data register [0A (hex) of the Index register]. Similar to the cursor-RAM write operation, when the read is completed the cursor-RAM address register is automatically incremented and further reads read successive cursor RAM locations.

The cursor RAM is written and read using the same hardware registers, so any task updating either of these on an interrupt thread must save and restore the cursor-RAM address LSB [Index 08 (hex)] and cursor-RAM address MSB [Index 09 (hex)] registers.

> **NOTES:** When the cursor-RAM address is to be written, always write both the cursor-RAM address LS and MS registers with the cursor-RAM address LSB register first.
>
> It is recommended that the cursor RAM not be accessed while the sprite cursor is enabled; otherwise, there is a possibility that the cursor RAM could be corrupted. Therefore, the sprite cursor should be temporarily disabled [cursor-control register (CCR) bit 6 = 0] when writing to or reading from the cursor RAM.
>
> The cursor-generation logic requires the use of active low sync inputs.
>
> Vertical retrace is determined by detecting multiple syncs in blank.
>
> The video front-porch time must be at least one RCLK period. The video back-porch time must be at least 80 pixel clock periods.

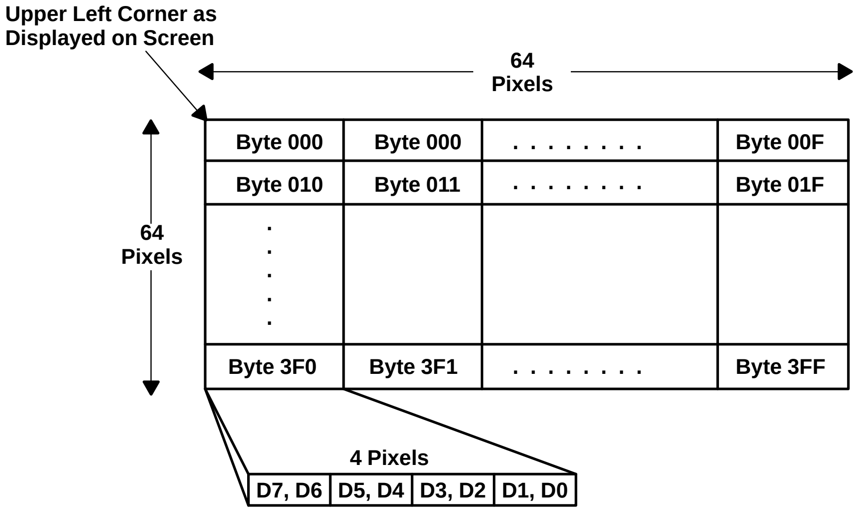


Figure 2–6. Cursor-RAM Organization

2.5.2 Two-Color 64 × 64 Cursor

The 64 × 64 × 2 cursor RAM provides two bits of cursor information on every pixel clock (DOTCLK) cycle during the 64 × 64 cursor window. Cursor control register bit 4 specifies whether the XGA mode (default = logic 0) or X-window mode (logic 1) standard is used to interpret the cursor information. The two bits of cursor pixel data determine the cursor appearance as shown in the table below:

RAM		COLOR SELECTION	
PLANE 1	PLANE 2	XGA MODE	X-WINDOW MODE
0	0	Cursor color 0	Transparent
0	1	Cursor color 1	Transparent
1	0	Transparent	Cursor color 0
1	1	Complement	Cursor color 1

Cursor color 0 and 1: These colors are set by writing to the cursor color 0 and cursor color 1 registers (index: 23–28 (hex)).

Transparent: The underlying pixel color is displayed.

Complement: The ones complement of the underlying pixel color is displayed.

2.5.3 64 × 64 Cursor Positioning

The cursor-position (x,y) registers are used in conjunction with the sprite-origin (x,y) registers to position the 64 × 64 cursor on the display screen. The cursor-position (x,y) registers specify the location of the cursor on the display screen relative to the first displayed pixel out of blank. The sprite-origin (x,y) register specifies where to origin the 64 x 64 cursor array relative to the cursor position (x,y). Upon reset, the sprite-origin (x,y) register automatically defaults to (31, 31). Therefore, the cursor-position (x,y) registers specify the location on the active display screen of the 31st column and 31st row (counting from top left) of the 64 x 64 cursor array. The crosshair cursor intersects at the center of the sprite cursor area.

The sprite-origin (x,y) registers can be programmed from (0,0) to (63,63). For example, if the sprite-origin (x,y) registers were programmed to (0,0), the 64 × 64 cursor array would be located in the lower right quadrant with respect to the cursor position (x,y). Figure 2–7 illustrates this more clearly by showing the 64 × 64 cursor array location relative to the cursor position (x) for different sprite-origin values.

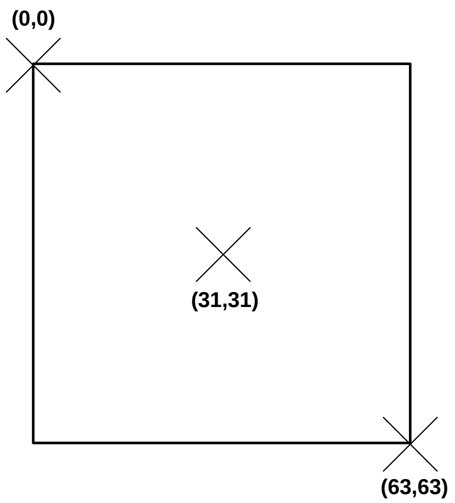


Figure 2–7. Common Sprite-Origin Settings

NOTE: The programmable sprite-origin feature can be especially useful in creating crosshair cursors and pointers. See Section 2.5.5 and Figures 2–8 and 2–9 for more details.

2.5.4 Crosshair Cursor

Cursor positioning for the crosshair cursor is also done through the cursor-position (x,y) register. The intersection of the crosshair cursor is specified by the cursor-position (x,y) register. If the thickness of the crosshair cursor is greater than one pixel, the center of the intersection is the reference position. The thickness of the crosshair cursor is specified by cursor-control register bits 0 and 1 (see Section 2.16.2). The sprite-origin (x,y) register has no effect on the crosshair cursor location.

In order to display the crosshair cursor, cursor-control register bit 2 needs to be enabled while CCR bit 3 is used to set the desired color as shown in the following table:

CCR2	CCR3	CROSSHAIR COLOR
0	0	Crosshair not displayed
0	1	Crosshair not displayed
1	0	Cursor color 0
1	1	Cursor color 1

Cursor-control register bits 0 and 1 specify the crosshair cursor thickness (see Section 2.16.2).

The crosshair cursor is limited to being displayed within a window, which is specified by the window-start (x,y) and window-stop (x, y) registers. Since the cursor-position (x,y) register must specify a point within the window boundaries, it is the responsibility of the user software to ensure that the cursor-position (x,y) register does not specify a point outside the defined window. Refer to Figure 2–8, which shows the relationship between the different window and cursor register specifying regions.

If a full-screen crosshair cursor is desired, the window-start (x,y) registers should contain 0000 (hex) and the window-stop (x,y) registers should be set to the last pixel location on the active screen. For the crosshair cursor to be displayed, the window-start and window-stop registers must contain locations on the active screen. If one wishes to temporarily remove the crosshair cursor from the screen without disabling the function, the window-start registers can be programmed with a location off the active screen.

The crosshair cursor and the auxiliary window function utilize the same set of window registers. Therefore, care must be taken if the crosshair cursor is to be displayed when the auxiliary window function is enabled (ACR0 = 1). See Section 2.6.

2.5.5 Dual-Cursor Positioning

Both the user-definable 64 × 64 cursor and the crosshair cursor may be enabled for display simultaneously, allowing the generation of custom crosshair cursors. As previously mentioned, the sprite-origin (x,y) register specifies the 64 × 64 cursor pattern location relative to the cursor position and crosshair cursor.

Figure 2–8 illustrates displaying the dual cursors, showing the relationship between the auxiliary window, the crosshair cursor, and the 64 × 64 cursor for the case where the sprite-origin (x,y) register is set to (31, 31).

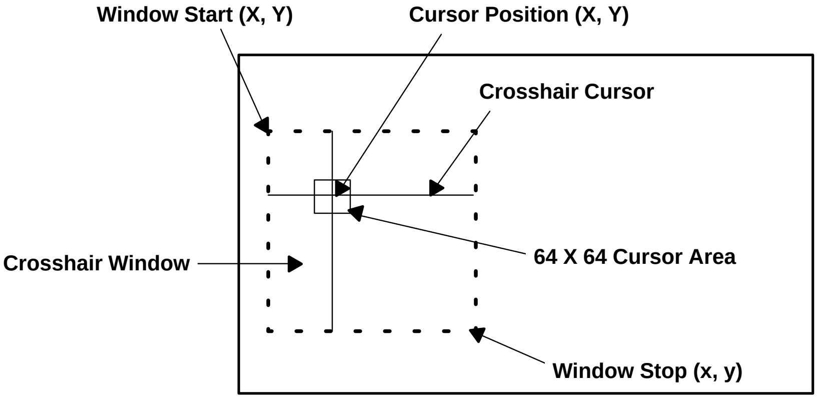


Figure 2–8. Dual-Cursor Positioning

Figure 2–9 shows one possible custom cursor that could be created by setting the sprite-origin register to (0,0) and drawing an arrow in 64 × 64 cursor RAM. The cursor window has been set to full screen by setting the window-start (x,y) register to 0000 (hex) and the window-stop registers to the last active pixel location. The 64 × 64 cursor area could be located in different locations about the cursor position by programming the sprite-origin (x,y) registers to different values as described earlier (see Section 2.5.3).

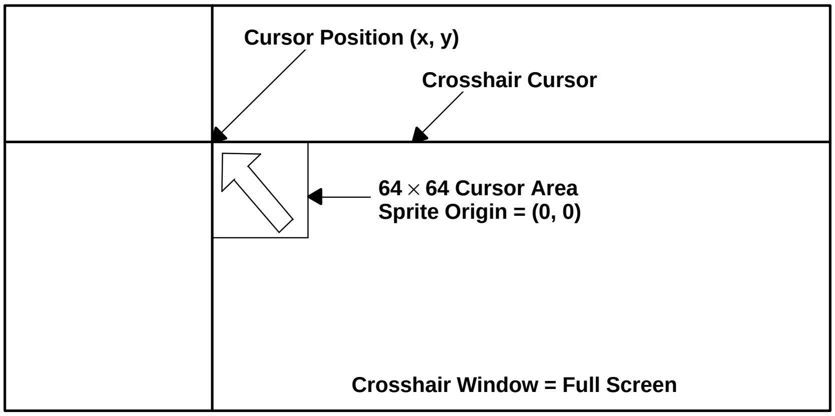


Figure 2–9. One Possible Custom Cursor Creation

When both the 64 x 64 user-definable cursor and the crosshair cursor are enabled, cursor-control register bit 5 specifies the display at the intersection of the crosshair cursor and the 64 × 64 user-definable cursor. The following cursor intersection truth table details the results of all cursor color combinations; see Section 2.16.2 for specific cursor-control register bit definitions.

CROSSHAIR	64 x 64 CURSOR	CCR5 = 0	CCR5 = 1
Color 0	Transparent	Color 0	Color 0
Color 1	Transparent	Color 1	Color 1
Color 0	Complement	Color 0	Color 0
Color 1	Complement	Color 1	Color 1
Color 0	Color 0	Color 0	Transparent
Color 1	Color 0	Color 1	Transparent
Color 0	Color 1	Color 0	Complement
Color 1	Color 1	Color 1	Complement

2.6 Auxiliary Window, Port Select, and Color-Key Switching

The Viewpoint palette provides three integrated mechanisms for switching between VGA or overlay images and direct-color images midscreen. The auxiliary window function supports the display of overlay or VGA graphics into a specified window on the screen when in a direct-color mode. The same window registers used to define the crosshair cursor window are used to define the auxiliary window start and window stop (see Sections 2.16.5 and 2.16.6). One application of this function is to fit a VGA picture in the middle of a direct-color display. The port-select function utilizes an external terminal (PSEL) to switch between VGA or overlay and direct-color on a pixel-by-pixel basis, enabling the generation of multiple VGA or overlay windows on a direct-color screen.

The auxiliary-window and port-select functions are integrated so that they can be enabled simultaneously or separately. They are only operable when in one of the direct-color modes, since both VGA and overlay utilize the palette RAM. Overlay windowing is not supported for those direct-color modes that do not have overlay capability. VGA windowing and VGA port selection are supported for all direct-color modes where the multiplex ratio is 1:1 (see Table 2 – 6). If VGA windowing is to be performed, multiplex-control register 2 needs to have bit 7 set to a logic 1 (activating the VGA port) and the appropriate direct-color mode must be chosen with the remaining multiplex-control register bits. When the VGA port is activated (MCR2 bit 7 = 1), the overlay is disabled and horizontal zooming is disabled.

The auxiliary-window and port-select functions are controlled by the auxiliary-control register, which is programmed through the indirect register map (29 hex, see Section 2.16.10 for register bit definitions).

Like the crosshair cursor window, for auxiliary graphics to be displayed, both the window-start and window-stop registers must contain locations on the active screen. If full-screen auxiliary graphics is desired, the window-start (x,y) registers should contain 0000 (hex) and the window-stop (x,y) registers should be set to the last active pixel location. The window-start register can be programmed with a location off the active screen to temporarily remove the auxiliary window without disabling the function entirely.

The color-key switching function allows switching between VGA or overlay and direct-color on a pixel-by-pixel basis by comparing the incoming VGA /overlay and direct-color data with user-programmable color-key ranges. The color-key ranges are set by writing to the eight 8-bit color-key range registers: color-key red (low, high), color-key green (low, high), color-key blue (low, high), and color-key OL / VGA (low, high). The color-key switching function is controlled by the color-key control register (see Section 2.6.2) All of the registers can be programmed through the indirect register map (see Sections 2.16.11 and 2.16.12 for register bit definitions). Color-key switching involving overlay is not supported for those direct-color modes that do not have overlay capability. Color-key switching involving VGA can be performed in all direct-color modes where the multiplex ratio is 1:1 (see Table 2 – 6). When the VGA port is activated (MCR2 bit 7 = 1), the OL/VGA register (low, high) color comparison is performed on VGA data and the VGA port is color-key switched instead of overlay.

The windowing and color-key switching functions are integrated much like a logical OR function. If either of the functions switches to palette graphics (VGA or overlay through the palette RAM), palette graphics are displayed instead of direct color. Therefore, when programming the device for any direct-color mode, both the color-key control and auxiliary-window registers must be set such that direct-color graphics is displayed. For true color (gamma corrected through the palette), one of the functions must be set to palette graphics.

All of the switching functions can be performed using self-clocked or externally clocked frame-buffer interface timing. Externally clocked timing allows all pixel port and VGA port timing to be referenced to CLK0 externally but can only be used for multiplex ratios of 1:1. If externally clocked timing is used, it is recommended that the VGA blank signal also be utilized. See Section 2.3.2 for specific details on clocking. All switching involving the VGA port can only be used with a 1:1 multiplex ratio.

2.6.1 Windowing Control

The Viewpoint palette supports several windowing formats. These are specified by the auxiliary-control-register bits 0–2 and PSEL as shown below.

Window context switching is determined by the following equation:

$$\text{SWITCH} = [(\text{PSEL} \times \text{ACR2}) + (\text{WINDOW} \times \text{ACR1})] \oplus \text{ACR0}$$

where:

WINDOW = 1 inside the auxiliary window.
ACR*n* is the nth bit of the auxiliary-control register.

The following table then applies:

MULTIPLEX MODE SELECTED (see Note 19)	DISPLAY RESULT	
	SWITCH = 0	**SWITCH = 1**
direct-color with VGA	direct-color	VGA
direct-color	direct-color	overlay

NOTES: 19. The multiplex mode is set by multiplex-control registers 1 and 2. If VGA switching is desired, multiplex-control register 2 bit 7 needs to be set to a logic 1 to enable the VGA port and the desired direct-color mode must be chosen with the remaining MCR bits. For example, if direct-color mode 1 is chosen and multiplex-control register 2 is normally set to 1B (hex), it would instead be set to 9B (hex) for VGA switching.

20. The DAC output is undefined if SWITCH = 1 when doing overlay switching in a direct-color mode that does not have overlay capability. If switching between direct color and VGA, any direct-color mode may be chosen as long as the multiplex ratio is 1:1.

21. Auxiliary-control register bits ACR2 and ACR1 can be used to independently enable or disable the port-select and windowing functions as shown in the equation above. If both switching functions are disabled, ACR0 is used to default the display to either direct color or palette graphics. Palette graphics are either VGA or overlay if in a direct-color mode or pseudo-color when in the pseudo-color mode. The RESET default is for palette graphics to be displayed – as needed for the VGA pass-through mode.

22. All of the switching modes that involve overlay and direct color support the multiple multiplexing ratios or LCLK divide ratios specified in Section 2.4.3 and Table 2–6 for those modes supporting overlay. However, caution must be observed when using the port-select function with the multiplexing modes other than 1:1 since the PSEL signal is latched on LCLK (same as the pixel port).

23. The windowing functions can be performed using self-clocked or externally clocked frame-buffer interface timing (see Section 2.3.2). If VGA switching is involved, CLK0 is the main clock source since VGA port data is latched on the rising edge of this signal. Self-clocked timing can be used by externally connecting RCLK to LCLK; however, this method is limited to a pixel rate of 50 MHz due to the delay from CLK0 to RCLK. Externally clocked timing references all pixel data latching to CLK0 by externally connecting CLK0 to LCLK. In both cases, the internal circuit pipeline delay is adjusted so that the VGA and pixel port data are synchronous in time.

The use of the auxiliary window to display a VGA window in a direct-color background can be accomplished by setting ACR2 = ACR0 = logic 0 and ACR1 = MCR2 bit 7 = logic 1 and is illustrated in Figure 2–10. The user can also configure the auxiliary window to display direct color in the auxiliary window and VGA everywhere else by setting ACR0 = logic 1 (not shown). Similarly, the auxiliary window can be configured to display overlay in the window or outside of the window by setting MCR2 bit 7 = logic 0 (not shown).

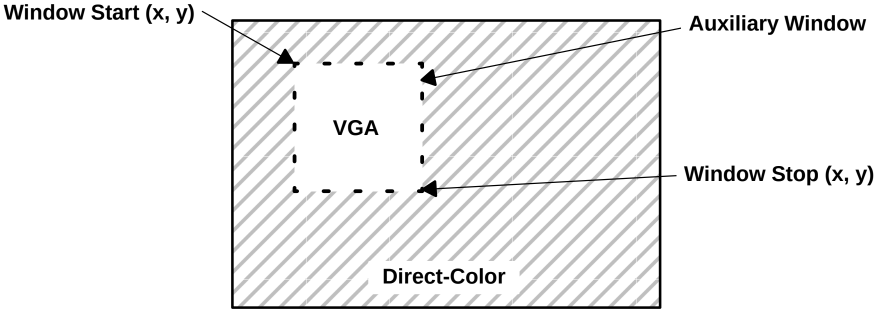


Figure 2–10. VGA in the Auxiliary Window

The use of PSEL to create multiple VGA windows in a direct-color background can be accomplished by setting ACR0 = ACR1 = logic 0 and ACR2 = logic 1. PSEL is then switched to logic 1 wherever VGA display is desired. This is illustrated in Figure 2–11. The user can also configure the port select to switch between overlay and direct color (MCR2 bit 7 = logic 0 not shown) and also invert the fields (ACR0 = logic 1).

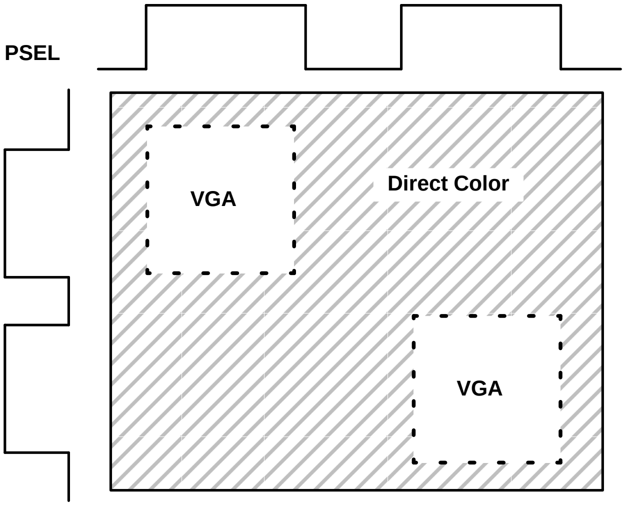


Figure 2–11. Multiple VGA Windows Using Port Select (PSEL)

2.6.2 Color-Key-Switching Control

The Viewpoint palette supports color-key-switching modes in which color data from the direct-color and overlay or VGA ports is compared to a set of user-definable color-key registers. Based on the outcome of the comparison, either direct color, overlay, or VGA are displayed (see Note 24). High and low color-key registers are provided for each color and overlay/VGA so that ranges of colors can be compared as opposed to a single color value. The register bit definitions for the color-key OL / VGA (low, high), color-key red (low, high), color-key green (low, high), and color-key blue (low, high) range registers are shown in Section 2.16.12. The color-key function is controlled by the color-key control register bits 0–4. This register definition is shown in Section 2.16.11.

Color-key switching is performed according to the following equation:

$$\text{COLOR}-\text{KEY} = [(\text{OL} + \overline{\text{CKC0}}) \times (\text{R} + \overline{\text{CKC1}}) \times (\text{G} + \overline{\text{CKC2}}) \times (\text{B} + \overline{\text{CKC3}})] \oplus \overline{\text{CKC4}}$$

where:	OL = 1	if	color-key OL / VGA low	≤ overlay or VGA (Note 24)	≤ color-key OL / VGA high
	R = 1	if	color-key red low	≤ direct color (RED)	≤ color-key red high
	G = 1	if	color-key green low	≤ direct color (GREEN)	≤ color-key green high
	B = 1	if	color-key blue low	≤ direct color (BLUE)	≤ color-key blue high

then if COLOR – KEY = 1, overlay or VGA is displayed.
if COLOR – KEY = 0, direct-color is displayed.

NOTES: 24. When the VGA port is activated (MCR2 bit 7 = 1), the OL / VGA register (low,high) color comparison is performed on VGA data and the VGA port is color-key switched. If the VGA port is not activated (MCR2 bit 7 = 0) the comparison is performed on overlay data and overlay is color-key switched.

25. Color-key switching is supported for all direct-color multiplexing modes that have overlay capability when doing overlay switching. When doing VGA switching, all direct-color modes are supported as long as the multiplex ratio is 1:1. The direct-color multiplex mode is set in multiplex-control registers 1 and 2.

26. CKC0 – CKC3 can be used to individually enable or disable certain colors in the comparison for maximum flexibility. If color-key switching is not desired, CKC0 – CKC3 should be set to logic 0. CKC4 is then used to set the default for either direct color or palette graphics. The default condition at RESET is CKC0 = CKC1 = CKC2 = CKC3 = logic 0 and CKC4 = logic 1. This causes the function to default to palette graphics as required for VGA pass-through node.

27. The color-key comparison for the overlay and VGA data is performed after the read mask and palette page registers so that an 8-bit comparison can be performed. This also gives the maximum flexibility to the user in performing the color comparisons. If the overlay defined for a given mode is less than 8 bits per pixel, the data is shifted to the LSB locations and the palette-page register fills the remaining MSB positions.

28. For those direct-color modes that have less than 8 bits per pixel of red, green, and blue direct-color data, the data is internally shifted to the MSB positions for each color and the remaining LSB bits are filled with logic 0s before the 8-bit comparisons are performed.

29. The windowing and color-key functions are integrated such that if either SWITCH = 1 (windowing case, see Section 2.6.1) or color key = 1, palette graphics are displayed (overlay or VGA depending on multiplex-control register 2 bit 7) instead of direct-color data. Both functions must be correctly set for proper operation.

2.7 Overscan

The Viewpoint palette provides the capability to produce a custom screen border using the overscan function. The overscan function is controlled by general-control-register (GCR) bits 6 and 7. GCR bit 6 is used to enable overscan, and GCR bit 7 specifies whether the overscan area is defined by OVS or by internal circuitry. The overscan color is user-programmable by writing to the overscan color red, green, and blue registers in the indirect register map.

If the overscan function is enabled (GCR6 = logic 1) and the OVS terminal is used to control the area of overscan (GCR7 = logic 0), then overscan color is displayed any time that OVS is high and $\overline{\text{BLANK}}$ is low (active). Note that $\overline{\text{BLANK}}$ is the internal blank signal and can either be generated from $\overline{\text{VGABL}}$ or $\overline{\text{SYSBL}}$ depending on the mode selected. If the Viewpoint internal circuitry is chosen to generate overscan (GCR7 = logic 1), then internal vertical and horizontal SYNC and BLANK are used to define the overscan display area. Whenever BLANK is active and vertical and horizontal SYNC are inactive, overscan is displayed. Internally generated timing may not work with some CRT monitors.

If overscan is enabled, then the blanking pedestal is imposed on the analog outputs when OVS is high and $\overline{\text{BLANK}}$ is low. If overscan is disabled, then the blanking pedestal occurs when $\overline{\text{BLANK}}$ is low. $\overline{\text{BLANK}}$ can be either $\overline{\text{SYSBL}}$ or $\overline{\text{VGABL}}$ depending on the state of multiplex-control register 2 bit 7.

If VGA is disabled, OVS is sampled on the falling edge of VCLK and then resampled on the rising edge of RCLK before being passed to the RCLK and dot clock pipeline delay. If VGA is enabled, then OVS is sampled on the rising edge of CLK0 and passed to dot clock pipeline delay. In this way, the video timing relationship is maintained since the same method and pipeline delay are applied to the $\overline{\text{SYSBL}}$ and $\overline{\text{VGABL}}$ signals.

Figure 2–12 demonstrates the use of OVS to produce a custom overscan screen border.

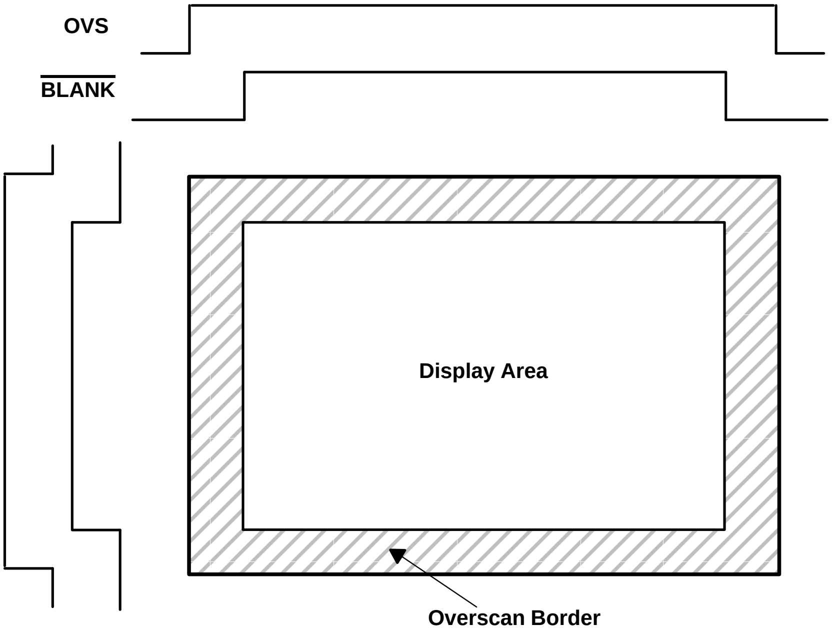


Figure 2–12. Overscan

2.8 Horizontal Zooming

The Viewpoint palette supports a user-programmable horizontal zooming function 2, 4, 8, 16, or 32x zoom. Zooming can be controlled through the auxiliary-control register on the indirect-register map as shown by the following table.

ACR7	ACR6	ACR5	HORIZONTAL ZOOM
0	0	0	1x
0	0	1	2x
0	1	0	4x
0	1	1	8x
1	0	0	16x
1	0	1	32x

When one of the horizontal zooms (besides 1x) is chosen, the dot clock is internally divided by the zoom number specified before it is allowed to clock the internal circuitry. Therefore, all internal circuitry is clocked at the divided frequency. The one exception to this is the VCLK generator, since the unmodified dot clock is sent to the VCLK counter circuitry. In this way, all of the video timing relationships are maintained.

The default at RESET is for the horizontal zoom to be 1x; i.e., disabled (ACR7 = ACR6 = ACR5 = logic 0). The programmed divide ratios for VCLK and RCLK/SCLK in the output-clock-selection register remain unchanged. Therefore, if horizontal zoom is chosen (other than 1x), SCLK is divided by the zoom number but VCLK remains unchanged.

If the divisor selected for VCLK in the output-clock-selection register is less than the zoom factor, VCLK defaults to the original dot frequency divided by the horizontal zoom factor. As an example of the zooming function, consider the case where dot clock is running at 100 MHz, RCLK/SCLK is programmed for dot/4, and VCLK is programmed for dot/2. In this case, RCLK/SCLK is running at 25 MHz and VCLK at 50 MHz.

Suppose that a horizontal zoom factor of 2X is then chosen. In that case, the internal dot clock (pixel rate) changes to 50 MHz, RCLK/SCLK changes to 12.5 MHz, and VCLK remains at 50 MHz. The effect on the screen is to display half as many pixels horizontally given the same video timing relationships.

It is recommended that the zoom only be changed during vertical retrace. Also, hardware cursor operation is not recommended.

The horizontal zoom function applies only to the pixel port (P0–63). If the VGA port is enabled (MCR2 bit 7 = 1), the horizontal zoom function is disabled.

2.9 Test Functions

The Viewpoint palette provides several functions that enable system testing and verification. These are detailed below.

2.9.1 16-Bit CRC

A 16-bit cyclic redundancy check (CRC) is provided so that video data integrity can be verified at the input to the DACs. The CRC is updated on the second horizontal sync (either $\overline{\text{SYSHS}}$ or VGAHS) rising edge during vertical retrace and is only calculated on the active screen area; i.e., active blank stops the calculation. The CRC can be performed on any of the 24 data lines that enter the DACs and is controlled by the CRC-control register (CRCC bits 0–4). Values from 0 to 23 may be written to this register to select between the 24 different DAC data inputs. Value 0 corresponds to DAC data red 0 (LSB), value 7 to red 7 (MSB), value 8 to green 0 (LSB), value 15 to green 7 (MSB), value 16 to blue 0 (LSB), and value 23 to blue 7 (MSB). The 16-bit remainder that is calculated on the individual DAC data line can be read from the CRCLSB and CRCMSB registers. See Table 2–2 for the indirect register map address. See Sections 2.16.14 and 2.16.15 for the CRC register bit definitions.

As long as the display pattern for each screen remains fixed, the CRC result should remain constant. If the CRC result changes, an error condition should be assumed. Since the CRC is calculated using the common CRC–16 polynomial ($X^{16} + X^{15} + X^{2} + 1$), the user can calculate and store the CRC remainder for a test screen in software and compare this to the Viewpoint-calculated CRC remainder to verify data integrity.

2.9.2 Sense Comparator Output and Test Register

The TVP3020 provides a $\overline{\text{SENSE}}$ output to support system diagnostics. $\overline{\text{SENSE}}$ can be used to determine the presence of the CRT monitor or verify that the RGB termination is correct. $\overline{\text{SENSE}}$ is a logic L if one or more of the DAC outputs exceeds the internal comparator voltage of 350 mV. The internal 350-mV reference has a tolerance of ±50 mV when using an external 1.235-V reference. If the internal voltage reference is used, the tolerance is higher.

The sense comparators are also integrated with the sense-test register so that the comparison results for the red, green, and blue comparators can be read independently through the 8-bit microinterface. When the sense-test register (STR) is read, the results are indicated in the bit positions as shown below.

STR BITS	D7	D6	D5	D4	D3	D2	D1	D0
Data	0	0	0	0	0	R	G	B

where: R = logic 1 if IOR > 350 mV   D6–D3 are reserved
G = logic 1 if IOG > 350 mV   D7 is disable (logic 1) bit
B = logic 1 if IOB > 350 mV

NOTES: A. D7 can be set to a logic 1 to disable the sense comparison function. At RESET, the sense comparison is enabled (D7 = logic 0). D6–D3 are reserved. If the sense-test register is written to disable the sense comparator function, bits D6–D0 need to be set to a logic 0.

B. Both the $\overline{\text{SENSE}}$ output and the sense-test register are latched by the falling edge of the internally sampled blank signal ($\overline{\text{SYSBL}}$ or $\overline{\text{VGABL}}$ depending on mode). In order to have stable voltage inputs to the comparators, the frame-buffer inputs should be set up such that data entering the DACs remains unchanged for a sufficient period of time prior to and after the blank-signal falling edge.

2.9.3 Identification Code

An ID register with a hardwired code is provided that can be used as a software verification for different versions of the system design. The ID code in the Viewpoint palette is static and may be read without consideration to the dot clock or video signals. The ID code is read through the indirect register map (see Table 2–2).

The value defined for the palette is 20 (hex).

2.10 General-Purpose I/O Register and Terminals

A general-purpose I/O register and output terminals are defined that provide a means of controlling external functions such as phase-locked loops (PLLs) through the Viewpoint microinterface. The 8-bit general-purpose I/O data register has five of its bit locations (D0 – D4) tied to external I/O terminals (I/O0 – I/O4). The other three bits (D5 – D7) can be used for general data storage and do not affect any other circuitry. The general-purpose I/O is controlled by the general-purpose I/O control register. G.P. I/O control register bits IOC0 – IOC4 control whether the corresponding general-purpose I/O terminals are configured as inputs or outputs. The reset default condition is for G.P. I/O control register bits IOC0 – IOC4 = logic 0, which configures terminals I/O0 – I/O4 as inputs. If any of the GPI/O control register bits are set to a logic 1, the corresponding I/O terminals are configured as outputs.

As mentioned before, this general-purpose I/O can be used to control external functions; i.e., serial programming of an external PLL through the Viewpoint microinterface.

The general-purpose I/O control register, data register, and terminal relationships are shown in the following table.

DATA BIT	D7	D6	D5	D4	D3	D2	D1	D0
General-Purpose I/O Control Register	X	X	X	IOC4	IOC3	IOC2	IOC1	IOC0
General-Purpose I/O Data Register	X	X	X	D4	D3	D2	D1	D0
General-Purpose I/O Pin Location				I/O4	I/O3	I/O2	I/O1	I/O0

2.11 Reset

There are two ways to reset the Viewpoint palette:

1. Hardware reset
2. Software reset

2.11.1 Hardware Reset

When the $\overline{\text{RESET}}$ input is held low, all Viewpoint registers go to the default state. This reset is asynchronous, and any glitch on this terminal could change the intended register setup. The default state at reset is VGA mode.

2.11.2 Software Reset

When the reset register [FF (hex) on the indirect register map] is written to, all other registers are initialized to VGA default settings accordingly. Any data may be written into the reset register to cause this reset to occur.

> **NOTE:** The default register settings are detailed in Tables 2–1 and 2–2.
>
> If reset is desired at power up, an external resistor, capacitor, and diode network should be connected to $\overline{\text{RESET}}$. See Figure A–1 for a typical circuit.
>
> If TTL logic is employed to provide the signal to the RESET terminal, a pullup resistor should be used.

2.12 Frame-Buffer Interface

The TVP3020 provides three output clock signals and one input clock signal for controlling the frame-buffer interface: SCLK, RCLK, LCLK, and VCLK. SCLK can be used to clock out data from VRAM shift registers directly. Split-shift register-transfer function is also supported. RCLK is provided so that pixel port (P0–P63) data loading can be synchronized to the VRAM. LCLK rising edges latch data presented on the pixel port, and VCLK is used to clock and synchronize the video control signals such as $\overline{\text{SYSHS}}$, $\overline{\text{SYSVS}}$, $\overline{\text{SYSBL}}$. Clocking of the frame-buffer interface (self-clocked and externally-clocked timing) is discussed in detail in Section 2.3.2.

The 64-terminal interface allows many operational display modes as defined in Section 2.4 and Table 2–6. The pixel latching sequence is initiated by a rising edge on LCLK. For those multiplexed modes in which multiple pixels are latched on one LCLK rising edge, the pixel clock shifts the pixels out starting with the pixels that reside on the low numbered pixel port terminals. For example, in an 8-bit-per-pixel pseudo-color mode with an 8:1 multiplex ratio, the pixel display sequence is P(0–7), P(8–15), P(16–23), P(24–31), P(32–39), P(40–47), P(48–55), and P(56–63).

The Viewpoint frame-buffer interface also supports little- and big-endian data formats on the pixel bus. This can be controlled by general-control register bit 3. See Sections 2.4.1 and 2.16.1, and Appendix C for details of operation.

2.13 Analog Output Specifications

The DAC outputs are controlled by three current sources (only two for IOR and IOB) as shown in Figure 2–13. The default condition is to have 0 IRE difference between blank and black levels, which is shown in Figure 2–15. If a 7.5-IRE pedestal is desired, it can be selected by setting bit 4 of the general-control register. This video output is shown in Figure 2–14.

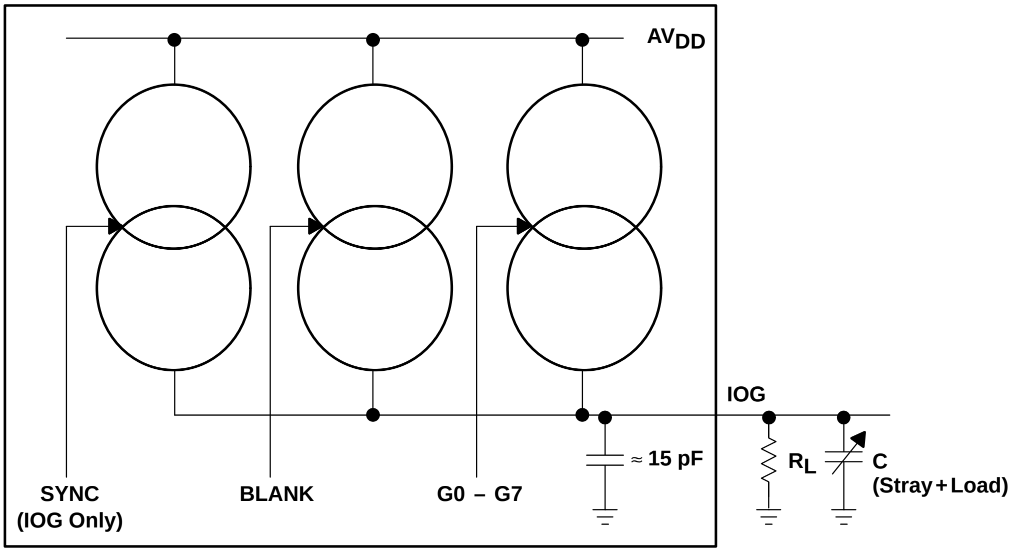


Figure 2–13. Equivalent Circuit of the Current Output (IOG)

A resistor (R_{SET}) is needed between the FS ADJ terminal and GND to control the magnitude of the full-scale video signal. The IRE relationships in Figures 2–14 and 2–15 are maintained regardless of the full-scale output current.

The relationship between RSET and the full scale output current IOG is:

$$\text{RSET } (\Omega) = \text{K1} \times V_{ref} \text{ (V) / IOG (mA)}$$

The full-scale output current on IOR and IOB for a given RSET is:

$$\text{IOR, IOB (mA)} = \text{K2} \times V_{ref} \text{ (V) / RSET } (\Omega)$$

where K1 and K2 are defined as:

PEDESTAL	IOG		IOR, IOB	
	8-BIT OUTPUT	6-BIT OUTPUT	8-BIT OUTPUT	6-BIT OUTPUT
7.5 IRE	K1 = 11,294	K1 = 11,206	K2 = 8,067	K2 = 7,979
0 IRE	K1 = 10,684	K1 =10,600	K2 = 7,462	K2 = 7,374

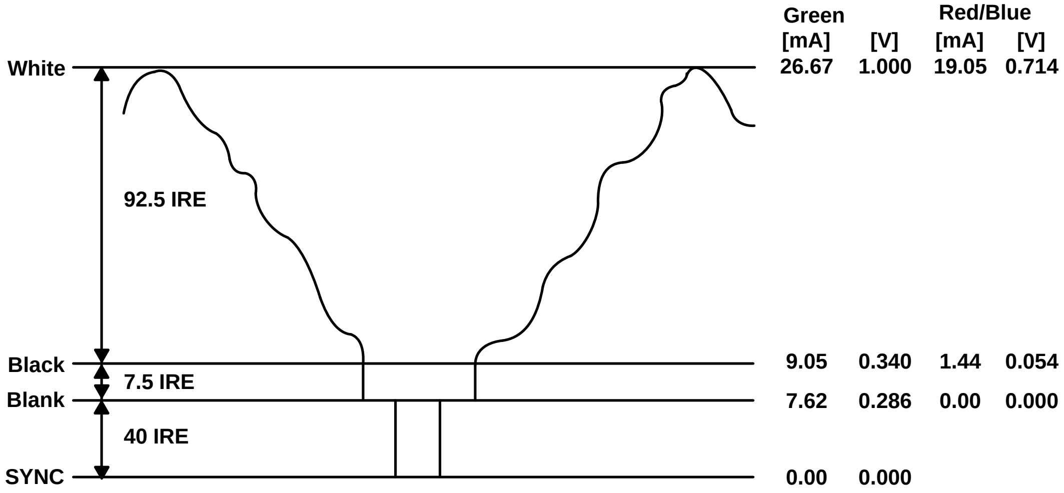


Figure 2–14. Composite Video Output (With 7.5 IRE, 8-Bit Output)

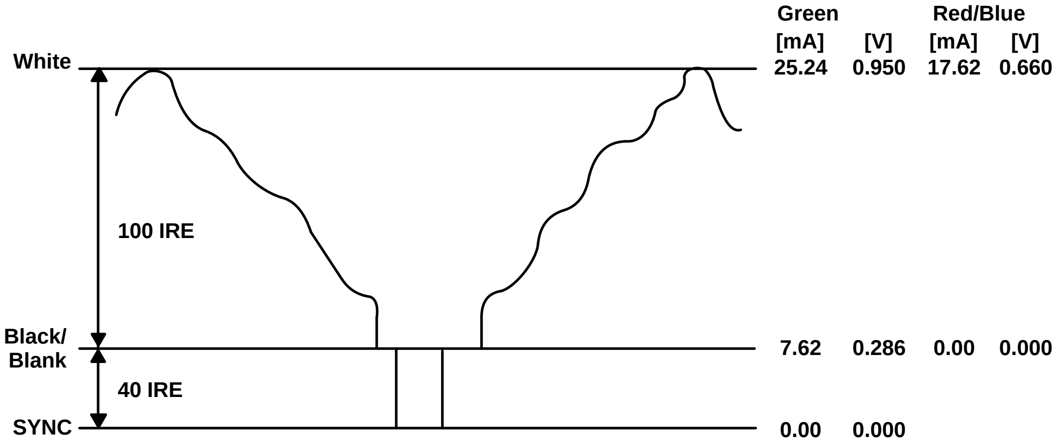


Figure 2–15. Composite Video Output (With 0 IRE, 8-Bit Output)

NOTE: 75-Ω doubly terminated load, V_{ref} = 1.235 V, RSET = 523 Ω. RS343A-levels and tolerances are assumed on all levels.

2.14 Video Control: Horizontal Sync, Vertical Sync, and Blank

For the high-resolution system modes, $\overline{\text{SYSHS}}$ and $\overline{\text{SYSVS}}$ are active-low pulses that are passed through true/complement gates to the HSYNCOUT and VSYNCOUT outputs. The output polarities of HSYNCOUT and VSYNCOUT can be programmed through the general-control register. However, for the VGA mode, the polarities required by the monitor are already provided at the feature connector where VGAHS and VGAVS are sourced. Therefore, the palette passes them through to HSYNCOUT and VSYNCOUT without polarity change. As described in Section 2.3 and Figures 2–2 through 2–5, the $\overline{\text{SYSBL}}$, $\overline{\text{SYSHS}}$ and $\overline{\text{SYSVS}}$ inputs are sampled and latched at the falling edge of VCLK in the system modes while $\overline{\text{VGABL}}$, VGAVS, and VGAHS are latched at the rising edge of CLK0 in the VGA mode. After $\overline{\text{SYSBL}}$ is sampled with VCLK, it is sampled on the rising edge of the internal RCLK and passed to the dot clock pipeline delay. When multiplex-control register 2 bit 7 is set to logic 1 to activate the VGA port, the CLK0, VGAHS, VGAVS, and $\overline{\text{VGABL}}$ inputs are selected. Otherwise, VCLK, $\overline{\text{SYSHS}}$, $\overline{\text{SYSVS}}$, and $\overline{\text{SYSBL}}$ are selected.

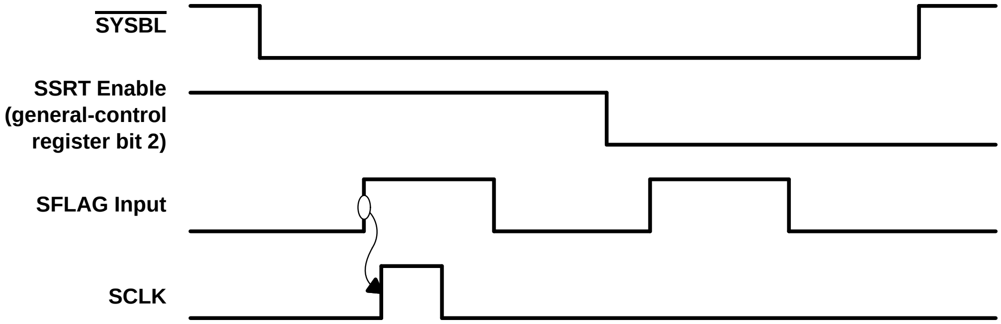
$\overline{\text{HSYNC}}$, $\overline{\text{VSYNC}}$, and $\overline{\text{BLANK}}$ (generated either from the system mode or VGA-mode video control signals) have internal pipeline delays so that the SYNC and BLANK signals align with the RGB data at the DAC outputs. Due to the sample and latch timing delay, it is possible to have active SCLKs after the selected blank input becomes active. The relationship between VCLK and SCLK and the internal VCLK sample and latch delay needs to be carefully reviewed and programmed. See Section 2.3 and Figures 2–2 and 2–3 for more details.

As shown in Figure 2–13, active $\overline{\text{HSYNC}}$ and $\overline{\text{VSYNC}}$ turns off the sync current source (after pipeline delay). They are not qualified by the $\overline{\text{BLANK}}$ signal. Therefore, to ensure proper operation, $\overline{\text{HSYNC}}$ and $\overline{\text{VSYNC}}$ should be designed such that they are active only during the blank active time.

To alter the polarity of the HSYNCOUT and VSYNCOUT outputs, the MPU must set or clear the corresponding bits in general-control register (see Section 2.16.1). The polarity of these signals can only be altered when not in VGA mode. These bits default to logic 0, which is an active-low output.

2.15 Split Shift-Register-Transfer VRAMs

The Viewpoint palette has direct support for split shift-register-transfer (SSRT) VRAMs. In order to allow the VRAMs to perform a split register transfer, an extra SCLK cycle must be inserted during the blank sequence. This is initiated when the SSRT enable bit (bit 2) in the general-control register is set to logic 1 and a rising edge on the SFLAG input is detected. An SCLK pulse is generated within 20 ns of the rising edge of the SFLAG signal. A minimum 15-ns logic-high duration is provided to satisfy all the –15 VRAM timing requirements. The rising edge of the SFLAG input triggers SCLK, but it needs to stay high for a specified minimum duration. By controlling the SFLAG timing, the delay time from the rising edge of VRAM $\overline{\text{TRG}}$ signal to SCLK can be satisfied. The relationship between SCLK, the SFLAG input, and $\overline{\text{SYSBL}}$ is shown as follows:



$\overline{\text{SYSBL}}$
SSRT Enable (general-control register bit 2)
SFLAG Input
SCLK

Figure 2–16. Split Shift-Register-Transfer Timing

If external SFLAG logic is designed as an R–S latch that is set by split shift-register-transfer timing and reset by $\overline{\text{SYSBL}}$ going high, the delay from $\overline{\text{SYSBL}}$ high to SFLAG low cannot exceed one half of one SCLK cycle. Otherwise, the SCLK generation logic could fail.

If the SSRT function is enabled but SFLAG is held low, the SCLK runs as if the SSRT function is disabled. Since the SFLAG input is not qualified by the $\overline{\text{BLANK}}$ signal within the palette, it needs to be held low or disabled any time the SSRT SCLK pulse is not intended. Refer to Section 2.3 and Figures 2–2 through 2–5 for more system details.

2.16 Control-Register Definitions

2.16.1 General-Control Register

The general-control register is used to control various functions of the Viewpoint palette. It can be accessed by the MPU at any time. Bit 7 of the general-control register corresponds to data bus bit 7, index = 1D (hex).

Table 2–12. General-Control Register

BIT NAME	VALUES	DESCRIPTION
GCR7	0: External OVS terminal (default)	Overscan-control select. Selects external terminal control or internally generated overscan control. See Section 2.7.
	1: Internal	
GCR6	0: Disable (default)	Overscan enable. Specifies whether to enable the user-defined overscan screen borders.
	1: Enable	
GCR5	0: Disable	Sync enable. This bit specifies whether SYNC information is to be output onto IOG.
	1: Enable (default)	
GCR4	0: 0 IRE (default)	Pedestal control. This bit specifies whether a 0 or 7.5 IRE blanking pedestal isto be generated on the video outputs. 0 IRE specifies that the black and blank levels are the same.
	1: 7.5 IRE	
GCR3	0: Little endian (default)	Little-endian/big-endian select. Selects either little- or big-endian format for the pixel-bus frame-buffer interface. See Sections 2.4 and Appendix C.
	1: Big-Endian	
GCR2	0: Disable (default)	Split shift-register-transfer enable. See Section 2.15.
	1: Enable	
GCR1	0: Active (low) (default)	VSYNCOUT output polarity. See Section 2.14.
	1: Active (high)	
GCR0	0: Active (low) (default)	HSYNCOUT output polarity. See Section 2.14.
	1: Active (high)	

2.16.2 Cursor-Control Register

The cursor-control register is used to control various on-chip cursor functions of the palette. It may be accessed by the MPU at any time. Bit 7 of the cursor-control register corresponds to data bus bit 7, index = 06 (hex).

Table 2–13. Cursor-Control Register

BIT NAME	VALUES	DESCRIPTION
CCR7	X	Reserved
CCR6	0: Disable (default)	Sprite-cursor enable. This bit enables (logic 1) or disables (logic 0) the 64 × 64 sprite cursor.
	1: Enable	
CCR5	0: (default)	Dual-cursor format. This bit specifies the display format at the intersection of the crosshair cursor and the user-defined cursor area. See Section 2.5.5 and the cursor intersection truth table.
	1:	
CCR4	0: XGA (default)	64 × 64 cursor-mode select. This bit specifies whether the XGA (logic 0) or X-windows format is used to interpret the data stored in the 64 × 64 cursor sprite RAM. See Section 2.5.2.
	1: X-windows	
CCR3	0: Color 0 (default)	Crosshair color selection. This bit specifies whether the crosshair cursor is to be displayed in color 1 (logical 1) or color 0 (logical 0).
	1: Color 1	
CCR2	0: Disable (default)	Crosshair cursor enable. This bit specifies whether the crosshair cursor is to be displayed (logical 1) or not (logical 0).
	1: Enable	
CCR1, CCR0	00: 1 pixel (default)	Crosshair thickness. These bits specify whether the vertical and horizontal thickness of the crosshair is one, three, five, or seven pixels. The segments are centered about the value in the cursor-position (x,y) register.
	01: 3 pixels	
	10: 5 pixels	
	11: 7 pixels	

2.16.3 Cursor-Position (x, y) Register

These registers are used to specify the (x,y) coordinate of the intersection of the crosshair cursor. They are also used in conjunction with the sprite-origin register to specify the location of the 64 x 64 cursor area (see Section 2.5.3). The cursor-position X register is made up of the cursor-position X LSB (CPXL) and the cursor-position X MSB (CPXM); the cursor-position Y register is made up of the cursor-position Y LSB (CPYL) and the cursor-position Y MSB (CPYM). All registers are initialized to 00 (hex) and can be written to or read from by the MPU at any time. The cursor position is not updated until the vertical retrace interval after CPYM has been written to by the MPU.

CPXL and CPXM are cascaded to form a 12-bit cursor-position X register. Similarly, CPYL and CPYM are cascaded to form a 12-bit cursor-position Y register. Bits D4–D7 of CPXM and CPYM are always a logic zero.

	CURSOR-POSITION X MSB								**CURSOR-POSITION X LSB**							
	(CPXM)								**(CPXL)**							
Data Bit	D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
X Position	0	0	0	0	X11	X10	X9	X8	X7	X6	X5	X4	X3	X2	X1	X0
	Index = 01h								Index = 00h							

	CURSOR-POSITION Y MSB								**CURSOR-POSITION Y LSB**							
	(CPYM)								**(CPYL)**							
Data Bit	D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
Y Position	0	0	0	0	Y11	Y10	Y9	Y8	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
	Index = 03h								Index = 02h							

The cursor-position X value to be written is calculated as follows:

Cx = desired display screen x position

Values from 0000 (hex) to 0fff (hex) may be written into the cursor-position X register.

The cursor-position Y value to be written is calculated as follows:

Cy = desired display screen y position

Values from 0000 (hex) to 0fff (hex) may be written into the cursor-position Y register.

The values written into the cursor-position X and Y registers should be relative to the first displayed pixel on the screen, i.e., (0,0).

2.16.4 Sprite-Origin (x, y) Register

These registers are used to specify the (x,y) location of the 64 x 64 sprite with respect to the crosshair location (see Section 2.5.3). The sprite-origin X and Y registers can contain values from 0 to 63 decimal. Both registers are initialized to 1F (hex), 31 (decimal), which sets the center of the crosshair at the center of the 64 X 64 sprite. Both registers may be written to or read from by the MPU at any time. The sprite origin is not updated until the vertical retrace interval after sprite-origin X and Y registers have been written by the MPU.

	SPRITE-ORIGIN X							
Data Bit	D7	D6	D5	D4	D3	D2	D1	D0
X Origin	0	0	X5	X4	X3	X2	X1	X0

Index = 04h

	SPRITE-ORIGIN Y							
Data Bit	D7	D6	D5	D4	D3	D2	D1	D0
Y Origin	0	0	Y5	Y4	Y3	Y2	Y1	Y0

Index = 05h

Values from 00 (hex) to 3F (hex) may be written into the sprite-origin X and Y registers. Bits D6 and D7 are always 0.

2.16.5 Window-Start (x, y) Register

These registers are used to specify the (x,y) coordinate of the upper-left corner of the crosshair-cursor window or auxiliary window. The window-start X register is made up of the window-start X LSB (WSXL) and the window-start X MSB (WSXM); the window-start Y register is made up of the window-start Y LSB (WSYL) and the window-start Y MSB (WSYM). They are not initialized and may be written to or read from by the MPU at any time. The window start is not updated until the vertical retrace interval after WSYM has been written to by the MPU.

WSXL and WSXM are cascaded to form a 12-bit window-start X register. Similarly, WSYL and WSYM are cascaded to form a 12-bit window-start Y register. Bits D4–D7 of WSXM and WSYM are always a logical zero.

	WINDOW START X MSB								**WINDOW START X LSB**							
	(WSXM)								**(WSXL)**							
Data Bit	D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
X Coordinate	0	0	0	0	X11	X10	X9	X8	X7	X6	X5	X4	X3	X2	X1	X0
	Index = 11h								Index = 10h							

	WINDOW START Y MSB								**WINDOW START Y LSB**							
	(WSYM)								**(WSYL)**							
Data Bit	D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
Y Coordinate	0	0	0	0	Y11	Y10	Y9	Y8	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
	Index = 15h								Index = 14h							

The window-start X value to be written is calculated as follows:

Wx = desired display screen x position

Values from 0000 (hex) to 0fff (hex) may be written into the window-start X register.

The window-start Y value to be written is calculated as follows:

Wy = desired display screen y position

Values from 0000 (hex) to 0fff (hex) may be written into the window-start Y register.

The values written into the window-start X and Y registers should be relative to the first displayed pixel on the screen; i.e., (0,0).

The window-start location specified is the first location inside the window. For the crosshair cursor or auxiliary window to be displayed, the window-start registers must specify a point on the active display.

2.16.6 Window-Stop (x, y) Register

These registers are used to specify the (x,y) coordinate of the lower-right corner of the crosshair cursor or auxiliary window. The window-stop X register is made up of the window-stop X LSB (WSPXL) and the window-stop X MSB (WSPXM); the window-stop Y register is made up of the window-stop Y LSB (WSPYL) and the window-stop Y MSB (WSPYM). They are not initialized and may be written to or read from by the MPU at any time. The window stop is not updated until the vertical retrace interval after WSPYM has been written to by the MPU.

WSPXL and WSPXM registers are cascaded to form a 12-bit window-stop X register. Similarly, WSPYL and WSPYM registers are cascaded to form a 12-bit window-stop Y register. Bits D4 – D7 of WSPXM and WSPYM are always a logical zero.

	WINDOW STOP X MSB								**WINDOW STOP X LSB**							
	(WSPXM)								**(WSPXL)**							
Data Bit	D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
X Coordinate	0	0	0	0	X11	X10	X9	X8	X7	X6	X5	X4	X3	X2	X1	X0

Index = 13h Index = 12h

	WINDOW STOP Y MSB								**WINDOW STOP Y LSB**							
	(WSPYM)								**(WSPYL)**							
Data Bit	D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
Y Coordinate	0	0	0	0	Y11	Y10	Y9	Y8	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0

Index = 17h Index = 16h

The window-stop X value to be written is calculated as follows:

Wx = desired display screen x position

Values from 0000 (hex) to 0fff (hex) may be written into the window-stop X register.

The window-stop Y value to be written is calculated as follows:

Wy = desired display screen y position

Values from 0000 (hex) to 0fff (hex) may be written into the window-stop Y register.

The values written into the window-stop X and Y registers should be relative to the first displayed pixel on the screen; i.e., (0,0).

The window-stop location specified is the last location inside the window. For the crosshair cursor or auxiliary window to be displayed, the window-start registers must specify a point on the active display.

2.16.7 Cursor-Color 0, 1 RGB Registers

These registers are used to specify the two colors for the hardware cursor (see Section 2.5). They are not initialized and may be written to or read from by the MPU at any time. Note that there are six registers total, three for each color. The register format for both the cursor-color 0 and cursor-color 1 registers is shown below.

	CURSOR COLOR RED							
Data Bit	D7	D6	D5	D4	D3	D2	D1	D0
Red Value	R7	R6	R5	R4	R3	R2	R1	R0

Index = 23h and 26h

	CURSOR COLOR GREEN							
Data Bit	D7	D6	D5	D4	D3	D2	D1	D0
Green Value	G7	G6	G5	G4	G3	G2	G1	G0

Index = 24h and 27h

	CURSOR COLOR BLUE							
Data Bit	D7	D6	D5	D4	D3	D2	D1	D0
Blue Value	B7	B6	B5	B4	B3	B2	B1	B0

Index = 25h and 28h

Values 00 (hex) to FF (hex) may be written into the cursor-color registers.

2.16.8 Cursor-RAM Address Register

These registers are used to specify the address of where to write or read sprite cursor data from. The linear addressing scheme is depicted in Figure 2–6 of Section 2.5.1. The cursor-RAM register is made up of the cursor-RAM address LSB (CRAL) and the cursor-RAM address MSB (CRAM). They are not initialized and may be written to by the MPU at any time.

CRAL and CRAM are cascaded to form a 10-bit cursor-RAM address register. Bits D2–D7 of CRAM are always a logical zero.

	CURSOR-RAM ADDRESS MSB								**CURSOR-RAM ADDRESS LSB**							
	(CRAM)								**(CRAL)**							
Data Bit	D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
Address	0	0	0	0	0	0	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

Index = 09h Index = 08h

Values from 0000 (hex) to 03FF (hex) may be written into the cursor-RAM address register.

When the cursor-RAM address is to be written, both registers must be written with the cursor-RAM address LSB being the first.

2.16.9 Cursor-RAM Data Register

This register is used to read and write the contents of the sprite cursor locations whose address is specified in the cursor-RAM address registers. The data read from or written to this register contain four pixels of information and two bit planes per cursor pixel (see Figure 2–6 and Section 2.5.1). The register is not initialized and may be written to or read from by the MPU at any time. The sprite-cursor data format is shown below.

	CURSOR-RAM DATA							
Data Bit	D7	D6	D5	D4	D3	D2	D1	D0
Data	$P1_3$	$P0_3$	$P1_2$	$P0_2$	$P1_1$	$P0_1$	$P1_0$	$P0_0$

Index = 0A h

Values from 00 (hex) to FF (hex) may be written into the cursor-RAM data register.

2.16.10 Auxiliary Control Register

The auxiliary control register is used to control various functions of the Viewpoint palette including the auxiliary-windowing function and horizontal zooming. It can be accessed by the MPU at any time. Bit 7 of the auxiliary-control register corresponds to data bus bit 7, index = 29 (hex).

BIT NAME	VALUES	DESCRIPTION
ACR7	0: (default)	Horizontal zoom control. See Section 2.8.
	1:	
ACR6	0: (default)	Horizontal zoom control. See Section 2.8.
	1:	
ACR5	0: (default)	Horizontal zoom control. See Section 2.8.
	1:	
ACR4	Reserved	
ACR3	0: Externally clocked	Frame-buffer clocking select. Used to select self- or externally-clocked timing for the pixel port (P0 – P63). See Section 2.3.2.
	1: Self clocked (default)	
ACR2	0: PSEL disable (default)	Windowing-function select. This is a port-select enable. See equation in Section 2.6.1.
	1: PSEL enable	
ACR1	0: Window disable (default)	Windowing-function select. This is an auxiliary-window enable. See the equation in Section 2.6.1.
	1: Window enable	
ACR0	0: True function	Windowing-function select. This is a complementary function bit. See the equation in Section 2.6.1.
	1: Complement (default)	

2.16.11 Color-Key-Control Register

The color-key-control register is used to control the operation of the color-key-switching function (see Section 2.6.2). It can be accessed by the MPU at any time. Bit 7 of the color-key-control register corresponds to data bus bit 7, index = 38 (hex).

BIT NAME	VALUES	DESCRIPTION
CKC7	Reserved	
CKC6	Reserved	
CKC5	Reserved	
CKC4	0: True function	Color-key-function select. This is a complementary function bit. See the equation in Section 2.6.2.
	1: Complement (default)	
CKC3	0: Disable compare (default)	Blue-compare enable. This is used to enable or disable the direct-color blue field comparison. See the equation in Section 2.6.2.
	1: Enable comparison	
CKC2	0: Disable compare (default)	Green-compare enable. This is used to enable or disable the direct-color green field comparison. See the equation in Section 2.6.2.
	1: Enable comparison	
CKC1	0: Disable compare (default)	Red-compare enable. This is used to enable or disable the direct-color red field comparison. See the equation in Section 2.6.2.
	1: Enable comparison	
CKC0	0: Disable compare (default)	Overlay/VGA-compare enable. This is used to enable or disable the direct-color overlay/VGA field comparison. See the equation in Section 2.6.2.
	1: Enable comparison	

2.16.12 Color-Key (Red, Green, Blue, Overlay) Low and High Registers

These registers are used to specify the color comparison ranges for the four direct-color data fields when performing color-key switching. A low and a high register are provided for each of the four data fields to facilitate the range comparison. See Sections 2.6 and 2.6.2 for more details on their usage. All four low registers are initialized with 01 (hex), while all four high registers are initialized with FF (hex). The registers may be written to or read from by the MPU at any time. There are eight registers total, two for each color. The register formats for both low and high registers are shown below.

	COLOR-KEY LOW							
Data Bit	D7	D6	D5	D4	D3	D2	D1	D0
Low Value	L7	L6	L5	L4	L3	L2	L1	L0

Index = 30h, 32h, 34h, and 36h

	COLOR-KEY HIGH							
Data Bit	D7	D6	D5	D4	D3	D2	D1	D0
High Value	H7	H6	H5	H4	H3	H2	H1	H0

Index = 31h, 33h, 35h, and 37h

Values 00 (hex) to FF (hex) may be written into the four color-key-low and four color-key-high registers.

2.16.13 Overscan-Color RGB Registers

These registers are used to specify the color for the overscan function. This function can be used to create custom screen borders (see Section 2.7). They are not initialized and may be written to or read from by the MPU at any time. The register formats for the overscan-color registers are shown below.

	OVERSCAN COLOR RED							
Data Bit	D7	D6	D5	D4	D3	D2	D1	D0
Red Value	R7	R6	R5	R4	R3	R2	R1	R0

Index = 20h

	OVERSCAN COLOR GREED							
Data Bit	D7	D6	D5	D4	D3	D2	D1	D0
Green Value	G7	G6	G5	G4	G3	G2	G1	G0

Index = 21h

	OVERSCAN COLOR BLUE							
Data Bit	D7	D6	D5	D4	D3	D2	D1	D0
Blue Value	B7	B6	B5	B4	B3	B2	B1	B0

Index = 22h

Values 00 (hex) to FF (hex) may be written into the overscan color registers.

2.16.14 CRC LSB and MSB Registers

These registers are used to read the result of the 16-bit CRC calculation (see Section 2.9.1). They are not initialized and may be read from by the MPU at any time. Note, however, that they are only updated on the rising edge of the second HSYNC during vertical retrace.

CRCLSB and CRCMSB are cascaded to form a 16-bit CRC calculation remainder.

	CRCMSB								**CRCLSB**							
Data Bit	D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
CRC Remainder	R15	R14	R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0
	Index = 3Dh								Index = 3Ch							

2.16.15 CRC-Control Register

This write-only register is used to specify which of the 24 DAC data lines the 16-bit CRC should be calculated on (see Section 2.9.1). The register is not initialized and may be written to by the MPU at any time. The CRC control register data format is shown below.

	CRC CONTROL REGISTER							
Data Bit	D7	D6	D5	D4	D3	D2	D1	D0
Select Data	0	0	0	D4	D3	D2	D1	D0
	Index = 3Eh							

Values from 00 (hex) to 17 (hex) may be written into the CRC control register.

3 Electrical Characteristics

3.1 Absolute Maximum Ratings Over Operating Free-Air Temperature Range (unless otherwise noted)†

Supply voltage, V_{DD} (see Note 1)	7 V
Input voltage range, V_I	–0.5 V to V_{DD} + 0.5 V
Analog output short-circuit duration to any power supply or common	unlimited
Operating free-air temperature, T_A	0°C to 70°C
Storage temperature range	–65°C to 150°C
Junction temperature	175°C
Case temperature for 10 seconds: FN package	260°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds	260°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTE 1: All voltage values are with respect to GND.

3.2 Recommended Operating Conditions

	MIN	NOM	MAX	UNIT
Supply voltages, $AV_{DD,}$ DV_{DD}	4.75	5	5.25	V
Reference voltage, V_{ref}	1.15	1.235	1.26	V
High-level input voltage, V_{IH}	2.4		V_{DD}+0.5	V
Low-level input voltage, V_{IL}			0.8	V
Output load resistance, R_L		37.5		Ω
FS ADJUST resistor, R_{SET}		523		Ω
Operating free-air temperature, T_A	0		70	°C

3.3 Electrical Characteristics

PARAMETER			TEST CONDITIONS	MIN	TYP†	MAX	UNIT
V_{OH}	High-level output voltage		I_{OH} = –800 µA	2.4			V
V_{OL}	Low-level output voltage	D<0:7>, I/O 0–4, VCLK, RCLK, $\overline{\text{SENSE}}$	I_{OL} = 3.2 mA			0.4	V
		HSYNCOUT, VSYNCOUT	I_{OL} = 15 mA			0.4	
		SCLK	I_{OL} = 18 mA			0.4	
I_{IH}	High-level input current	TTL inputs	V_I = 2.4 V			1	µA
		ECL inputs	V_I = 4 V			1	
I_{IL}	Low-level input current	TTL inputs	V_I = 0.8 V			–1	µA
		ECL inputs	V_I = 0.4 V			–1	
I_{DD}	Supply current, pseudo-color mode (see Note 2)	TVP3020-135	V_{DD} = 5		350	500	mA
		TVP3020-175			450	575	
		TVP3020-200			500	650	
	Supply current, true-color mode	TVP3020-135			400	500	
		TVP3020-175			500	575	
		TVP3020-200			550	650	
I_{OZ}	High-impedance-state output current					10	µA
C_i	Input capacitance	TTL inputs	f = 1 MHz, V_I = 2.4 V		4		pF
		ECL inputs	f = 1 MHz, V_I = 4 V		4		
V_{ID}	Differential input voltage	ECL inputs		0.6		6	V
V_{IC}	Common-mode input voltage	ECL inputs		2.85	3.15	V_{DD} – 0.5	V

† All typical values are at V_{DD} = 5 V, T_A = 25°C.

NOTE 2: I_{DD} is measured with DOTCLK running at the maximum specified frequency, SCLK frequency = DOTCLK frequency/8, and the palette RAM loaded with repeating full-range toggling patterns (00h/00h/00h/00h/FFh/FFh/FFh/FFh). Pseudo-color mode is also known as color indexing mode.

3.4 Operating Characteristics

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
	Resolution (each DAC)	$8/\overline{6}$ high		8		bits
		$8/\overline{6}$ low		6		
E_L	End-point linearity error (each DAC)	$8/\overline{6}$ high			1	LSB
		$8/\overline{6}$ low			1/4	
E_D	Differential linearity error (each DAC)	$8/\overline{6}$ high			1	LSB
		$8/\overline{6}$ low			1/4	
	Gray scale error				5%	
	Output current (see Note 3)	White level relative to blank	17.69	19.05	20.4	mA
		White level relative to black (7.5 IRE only)	16.74	17.62	18.5	mA
		Black level relative to blank (7.5 IRE only)	0.95	1.44	1.9	mA
		Blank level on IOR, IOB	0	5	50	µA
		Blank level on IOG (with SYNC enabled)	6.29	7.6	8.96	mA
		Sync level on IOG (with SYNC enabled)	0	5	50	µA
		One LSB ($8/\overline{6}$ high)		69.1		µA
		One LSB ($8/\overline{6}$ low)		276.4		µA
	DAC-to-DAC matching			2%	5%	
	DAC-to-DAC crosstalk			–20		dB
	Output compliance		–1		1.2	V
	Voltage reference output voltage		1.15	1.235	1.26	V
	Output impedance			50		kΩ
	Output capacitance	f = 1 MHz, $I_{OUT} = 0$		13		pF
	Sense voltage reference		300	350	400	mV
	Clock and data feedthrough			–20		dB
	Glitch impulse (see Note 4)			50		pV–s
	Pipeline delay, VGA port	Self clocked timing		18 DOT		periods
		Externally clocked timing		15 DOT		periods
	Pipeline delay, pixel port	Self clocked timing		2 RCLK + 14 DOT		periods
		Externally clocked timing		1 RCLK +14 DOT		periods

NOTES: 3. Test conditions for RS343-A video signals (unless otherwise specified): “Recommended Operating Conditions”, using external voltage reference $V_{ref} = 1.235$ V, $R_{SET} = 523\ \Omega$. When using the internal voltage reference, R_{SET} may need to be adjusted in order to meet these limits.

4. Glitch impulse does not include clock and data feedthrough. The – 3-dB test bandwidth is twice the clock rate.

3.5 Timing Requirements (see Note 5)

			TVP3020 -135		TVP3020 -175		TVP3020 -200		UNIT
			MIN	MAX	MIN	MAX	MIN	MAX	
	DOTCLK frequency			135		170		200	MHz
	CLK0 frequency for VGA pass-through mode (see Note 6)			85		85		85	MHz
t_{cyc}	Clock cycle time	TTL	7.4		7.1		7.1		ns
		ECL	7.4		5.8		5.0		
t_{su1}	Setup time, RS(0 – 3) valid before $\overline{\text{RD}}$ or $\overline{\text{WR}}$↓		10		10		10		ns
t_{h1}	Hold time, RS(0 – 3) valid after $\overline{\text{RD}}$ or $\overline{\text{WR}}$↓		10		10		10		ns
t_{su2}	Setup time, D(0 – 7) valid before $\overline{\text{WR}}$↑		35		35		35		ns
t_{h2}	Hold time, D(0 – 7) valid after $\overline{\text{WR}}$↑		0		0		0		ns
t_{su3}	Setup time, VGA(0 – 7) and VGAHS, VGAVS, and $\overline{\text{VGABL}}$ valid before CLK0↑ (see Note 7)		2		2		2		ns
t_{h3}	Hold time, VGA(0 – 7) and VGAHS, VGAVS, and $\overline{\text{VGABL}}$ valid after CLK0↑ (see Note 7)		2		2		2		ns
t_{su4}	Setup time, P(0 – 63) and PSEL valid before LCLK↑ (see Note 8)		2		2		2		ns
t_{h4}	Hold time, P(0 – 63) and PSEL valid after LCLK↑ (see Note 8)		5		5		5		ns
t_{su5}	Setup time, $\overline{\text{SYSHS}}$, $\overline{\text{SYSVS}}$, and $\overline{\text{SYSBL}}$ valid before VCLK↓		5		5		5		ns
t_{h5}	Hold time, $\overline{\text{SYSHS}}$, $\overline{\text{SYSVS}}$, and $\overline{\text{SYSBL}}$ valid after VCLK↓		1		1		1		ns
t_{w1}	Pulse duration, $\overline{\text{RD}}$ or $\overline{\text{WR}}$ low		50		50		50		ns
t_{w2}	Pulse duration, $\overline{\text{RD}}$ or $\overline{\text{WR}}$ high		30		30		30		ns
t_{w3}	Pulse duration, clock high	TTL	3		3		3		ns
		ECL	3		2.5		2		
t_{w4}	Pulse duration, clock low	TTL	3		3		3		ns
		ECL	3		2.5		2		
t_{w5}	Pulse duration, SFLAG high (see Note 9)		30		30		30		ns
t_{w6}	Pulse duration, SCLK high (see Note 9)		15	55	15	55	15	55	ns

NOTES: 5. TTL input signals are 0 to 3 V with less than 3 ns rise/fall time between the 10% and 90% levels unless otherwise specified. ECL input signals are V_{DD}–1.8 V to V_{DD}–0.8 V with less than 2 ns rise/fall time between the 20% and 80% levels. For input and output signals, timing reference points are at the 10% and 90% signal levels. Analog output loads are less than 10 pF. D<0:7> output loads are less than 50 pF. All other output loads are less than 50 pF unless otherwise specified.

6. In VGA mode, CLK0 minimum pulse duration for clock low should be greater than 4.8 ns. If VGA switching is to be performed using self-clocked timing, the maximum pixel rate cannot exceed 50 MHz.

7. Reference to CLK0 input only.

8. RCLK is delayed from SCLK in such a way that when RCLK is connected to LCLK, the timing is essentially the same as the TLC3407x family of parts.

9. This parameter applies when the split shift register transfer (SSRT) function is enabled. See Section 2.15 for details.

3.6 Switching Characteristics

PARAMETER		TVP3020-135			UNIT
		MIN	TYP	MAX	
	SCLK frequency (C_{LOAD} ≤ 15 pF) (see Note 10)			85	MHz
	SCLK frequency (C_{LOAD} ≤ 60 pF) (see Note 10)			85	MHz
	RCLK/VCLK frequency (see Note 10)			85	MHz
t_{en1}	Enable time, $\overline{RD}$ low to D(0 – 7) valid			40	ns
t_{dis1}	Disable time, $\overline{RD}$ high to D(0 – 7) disabled			17	ns
t_{v1}	Valid time, D(0 – 7) valid after $\overline{RD}$ high	5			ns
t_{PLH1}	Propagation delay, SFLAG↑ to SCLK high (see Note 10 and 11)	0		20	ns
t_{d1}	Delay time, $\overline{RD}$ low to D(0 – 7) starting to turn on	5			ns
t_{d2}	Delay time, selected input clock high/low to DOTCLK (internal signal) high/low		7		ns
t_{d3}	Delay time, SCLK high/low to RCLK high/low (see Note 12)	1	2	5	ns
t_{d4}	Delay time, VCLK high/low to RCLK high/low (see Note 12)	1	3	6	ns
t_{d5}	Delay time, RCLK high/low from DOTCLK high/low (internal signal)		7		ns
t_{d6}	Delay time, LCLK from RCLK			t_{RCLK-7}	ns
t_{d7}	Delay time, DOTCLK high to IOR/IOG/IOB active (analog output delay time) (see Note 13)		4		ns
t_{d8}	Analog output settling time (see Note 14)		6		ns
t_{d9}	Delay time, DOTCLK high to HSYNCOUT and VSYNCOUT valid		9		ns
t_r	Analog output rise time (see Note 15)		2		ns
	Analog output skew	0		2	ns

NOTES: 10. SCLK can drive an output capacitive load up to 60 pF. The worst-case transition time between the 10% and 90% levels is less than 4 ns (typical 3 ns). RCLK and VCLK can drive output capacitive loads up to 15 pF, with worst case transition times between 10% and 90% levels less than 4 ns (typical 3 ns).

11. This parameter applies when the split shift register transfer (SSRT) function is enabled. See Section 2.15 for details.

12. The SCLK and VCLK delay time to RCLK depends on the load that the signals drive. This parameter is measured with an RCLK to VCLK ratio of 1:1, and VCLK = RCLK load of 15 pF and SCLK load of 60 pF.

13. Measured from the 90% point of the rising edge of DOTCLK to 50% of the full-scale transition.

14. Measured from the 50% point of the full-scale transition to the point at which the output has settled, within ± 1 LSB (settling time does not include clock and data feedthrough).

15. Measured between 10% and 90% of the full-scale transition.

3.6 Switching Characteristics (Continued)

PARAMETER		TVP3020-175			TVP3020-200			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	
	SCLK frequency (C_{LOAD} ≤ 15 pF) (see Note 10)			87.5			100	MHz
	SCLK frequency (C_{LOAD} ≤ 60 pF) (see Note 10)			85			85	MHz
	RCLK, VCLK frequency (see Note 10)			87.5			100	MHz
t_{en1}	Enable time, $\overline{RD}$ low to D(0 – 7) valid			40			40	ns
t_{dis1}	Disable time, $\overline{RD}$ high to D(0 – 7) disabled			17			17	ns
t_{v1}	Valid time, D(0 – 7) valid after $\overline{RD}$ high	5			5			ns
t_{PLH1}	Propagation delay, SFLAG↑ to SCLK high (see Note 10 and 11)	0		20	0		20	ns
t_{d1}	Delay time, $\overline{RD}$ low to D(0 – 7) starting to turn on	5			5			ns
t_{d2}	Delay time, selected input clock high/low to DOTCLK (internal signal) high/low		7			7		ns
t_{d3}	Delay time, SCLK high/low to RCLK high/low (see Note 12)	1	2	5	1	2	5	ns
t_{d4}	Delay time, VCLK high/low to RCLK high/low (see Note 12)	1	3	6	1	3	6	ns
t_{d5}	Delay time, RCLK high/low from DOTCLK high/low (internal signal)		7			7		ns
t_{d6}	Delay time, LCLK from RCLK			t_{RCLK-7}			t_{RCLK-7}	ns
t_{d7}	Delay time, DOTCLK high to IOR/IOG/IOB active (analog output delay time) (see Note 13)		4			4		ns
t_{d8}	Analog output settling time (see Note 14)		5			5		ns
t_{d9}	Delay time, DOTCLK high to HSYNCOUT and VSYNCOUT valid		9			9		ns
t_r	Analog output rise time (see Note 15)		2			2		ns
	Analog output skew	0		2	0		2	ns

NOTES: 10. SCLK can drive an output capacitive load up to 60 pF. The worst-case transition time between the 10% and 90% levels is less than 4 ns (typical 3 ns). RCLK and VCLK can drive output capacitive loads up to 15 pF, with worst case transition times between 10% and 90% levels less than 4 ns (typical 3 ns).

11. This parameter applies when the split shift register transfer (SSRT) function is enabled. See Section 2.15 for details.

12. The SCLK and VCLK delay time to RCLK depends on the load that the signals drive. This parameter is measured with an RCLK to VCLK ratio of 1:1, and VCLK = RCLK load of 15 pF and SCLK load of 60 pF.

13. Measured from the 90% point of the rising edge of DOTCLK to 50% of the full-scale transition.

14. Measured from the 50% point of the full-scale transition to the point at which the output has settled, within ± 1 LSB (settling time does not include clock and data feedthrough).

15. Measured between 10% and 90% of the full-scale transition.

3.7 Timing Diagrams

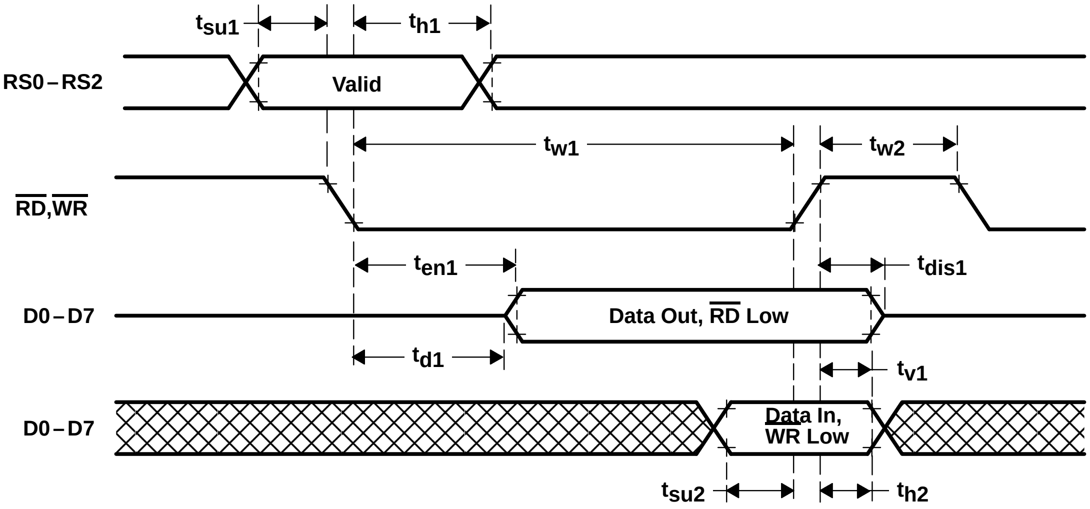


Figure 3–1. MPU Interface Timing

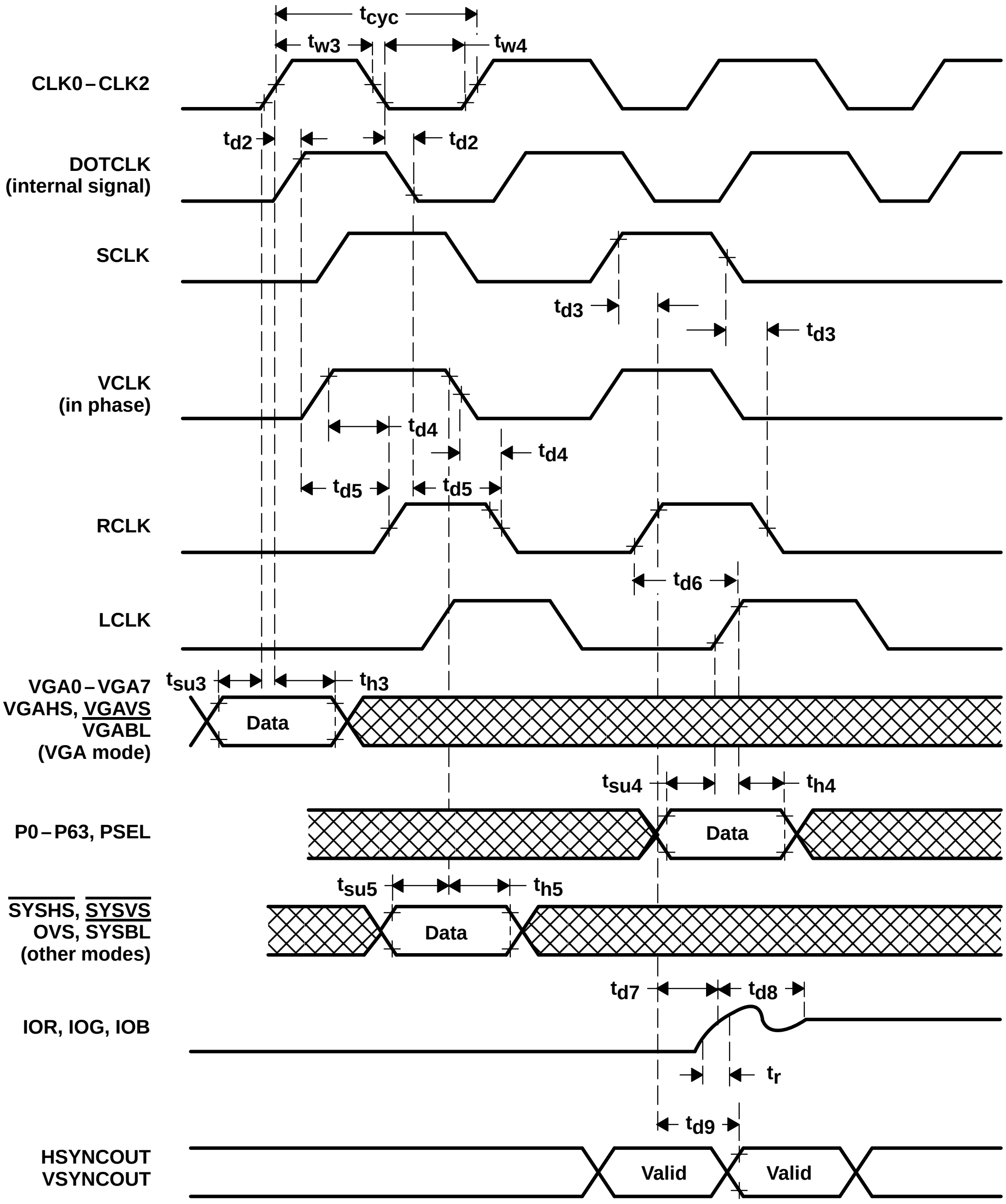


Figure 3–2. Video Input/Output Timing

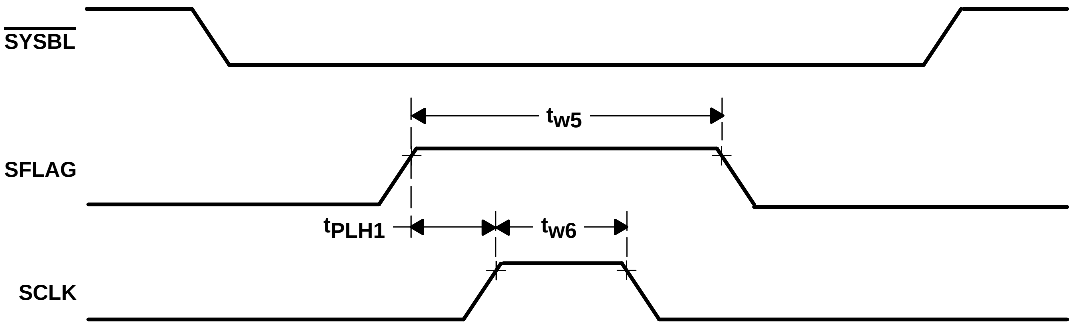


Figure 3–3. SFLAG Timing (When SSRT Function is Enabled)

Appendix A
PC-Board Layout Considerations

PC-Board Considerations

It is recommended that a four-layer PC board be used with the Viewpoint video interface palette: one layer for 5-V power, one for GND, and two for signals. The layout should be optimized for the lowest noise on the Viewpoint power and ground lines by shielding the digital inputs and providing good decoupling. The lead length between groups of analog V_{DD} and GND terminals (see Figure A–1) should be minimized so as to minimize inductive ringing. The TVP3020 P0–P64 terminal assignments have been selected for minimum interconnect lengths between these inputs and the standard VRAM pixel data outputs. The TVP3020 should be located as close as possible to the output connectors to minimize noise pickup and reflections due to impedance mismatch.

The analog outputs are susceptible to crosstalk from digital lines; digital traces must not be routed under or adjacent to the analog output traces.

For maximum performance, the analog-video-output impedance, cable impedance, and load impedance should be the same. The load resistor connection between the video outputs and GND should be as close as possible to the TVP3020 to minimize reflections. Unused analog outputs should be connected to GND.

Analog output video edges exceeding the CRT monitor bandwidth can be reflected, producing cable-length-dependent ghosts. Simple pulse filters can reduce high-frequency energy, thus reducing EMI and noise. The filter impedance must match the line impedance.

Ground Plane

It is also recommended that only one ground plane be used for both the TVP3020 and the rest of the logic. Separate digital and analog ground planes are not needed and can potentially cause system problems.

Power Plane

Split-power planes for the TVP3020 and the rest of the logic are recommended. The Viewpoint VIP and its associated analog circuitry should have its own power plane, referred to as analog V_{DD}. These two power planes should be connected at a single point through a ferrite bead, as shown in Figures A–1, A–2, and A–3.This bead should be located as near as possible to where the power supply connects to the board. To maximize the high-frequency power supply rejection, the video output signals should not overlay the analog power plane.

Supply Decoupling

The bypass capacitors should be installed using the shortest leads possible. This reduces the lead inductance and is consistent with reliable operation.

For the best performance, a 0.1-µF ceramic capacitor in parallel with a 0.01-µF chip capacitor should be used to decouple each of the groups of power terminals to GND. These capacitors should be placed as close as possible to the device, as shown in Figure A–2.

If a switching power supply is used, the designer should pay close attention to reducing power supply noise and consider using a three-terminal voltage regulator for supplying power to AV_{DD}.

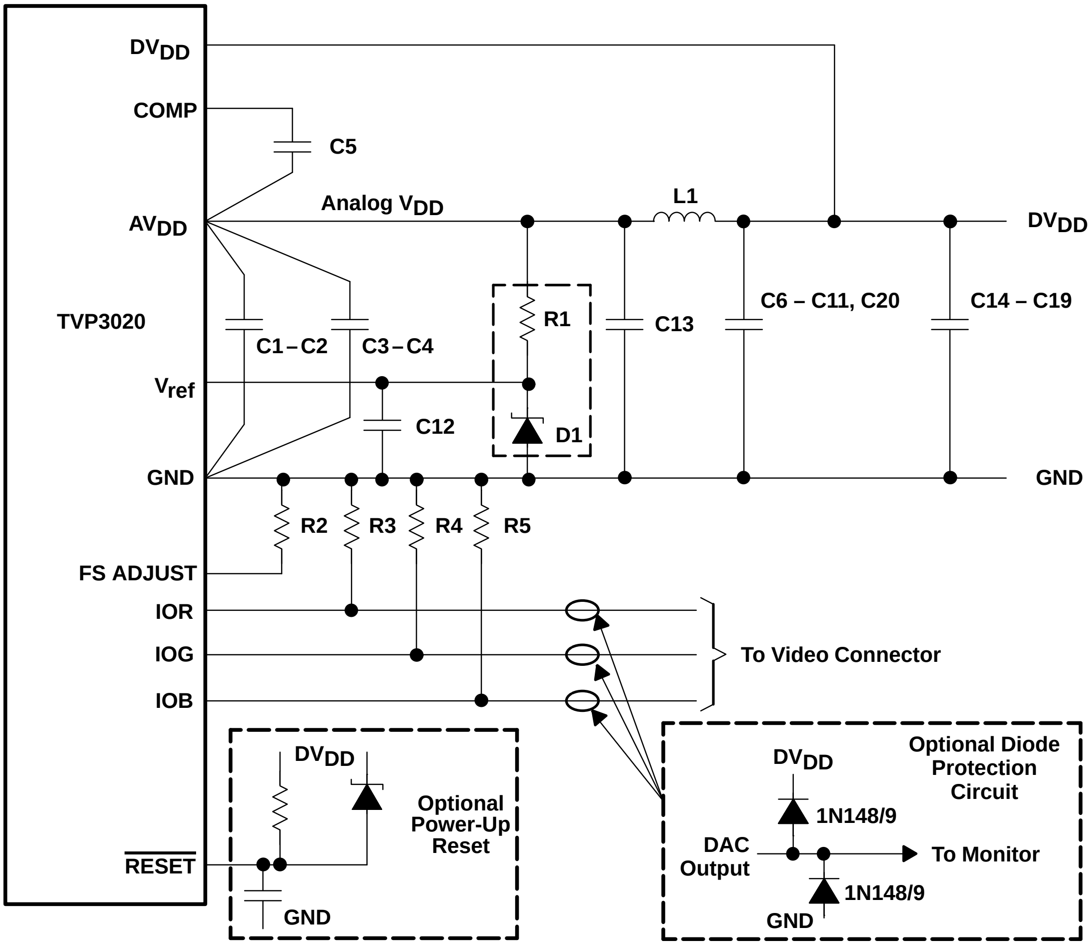
COMP and REF Pins

A 0.1-μF ceramic capacitor should be connected between AV_{DD} and COMP to avoid noise and color-smearing problems. A 0.1-μF ceramic capacitor is also recommended between GND and REF to further stabilize the output image. This 0.1-μF capacitor is needed for either internal or external voltage references. These capacitor values may depend on the board layout; experimentation may be required in order to determine optimum values.

Analog Output Protection

The TVP3020 analog output should be protected against high-energy discharges, such as those from monitor arc-over or from hot-switching ac-coupled monitors.

The diode protection circuit shown in Figure A–1 can prevent latch-up under severe discharge conditions without adversely degrading analog transition times. The IN4148/9 parts are low-capacitance, fast-switching diodes, which are also available in multiple-device packages (FSA250X or FSA270X) or surface-mountable pairs (BAV99 or MMBD7001).



Location	Description	VENDOR PART NUMBER[†]
C1, C2, C5 – C12, C20	0.1-µF ceramic capacitor	Erie RPE110Z5U104M50V
C3, C4, C14 – C19	0.01-µF ceramic chip capacitor	AVX 12102T103QA1018
C13	33-µF tantalum capacitor	Mallory CSR13F336KM
L1	Ferrite bead	Fair-Rite 2743001111[‡]
R1	1000-Ω 1% metal-film resistor	Dale CMF-55C
R2	523-Ω 1% metal-film resistor	Dale CMF-55C
R3, R4, R5	75-Ω 1% metal-film resistor	Dale CMF-55C
D1	1.2-V voltage reference	TI LM385-1.2

[†] The vendor numbers above are listed only as a guide. Substitution of devices with similar characteristics will not affect the performance of the TVP3020.

[‡] Or equivalent only.

Figure A–1. Typical Connection Diagram and Parts

NOTE: R1, D1, and reset circuit are optional. In general, each pair of device power and GND pins should be separately decoupled with 0.1-µF and 0.01-µF capacitors.

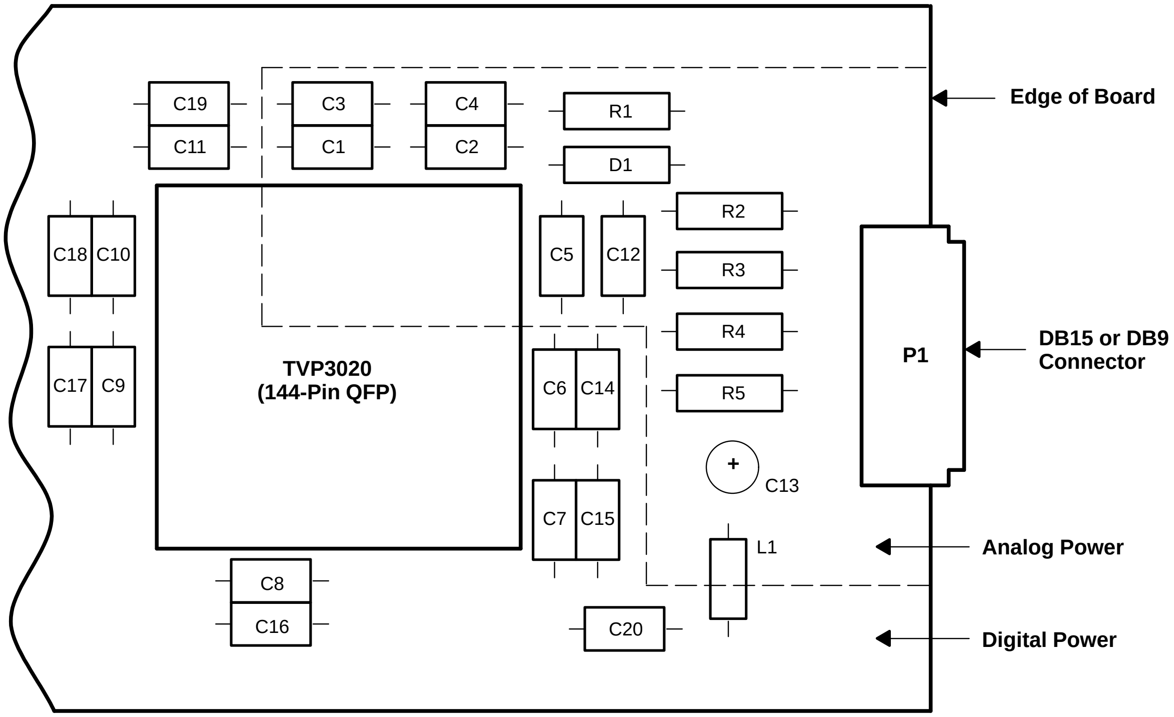


Figure A–2. Typical Component Placement With Split-Power Plane

Appendix B
RCLK Frequency < VCLK Frequency

The VCLK, RCLK, and SCLK outputs generated by the Viewpoint VIP are free-running clocks. The video-control signals (i.e., $\overline{\text{HSYNC}}$, $\overline{\text{VSYNC}}$, and $\overline{\text{BLANK}}$) are normally generated from VCLK, and a fixed relationship between video-control signals and VCLK can be expected. Viewpoint samples and latches the $\overline{\text{BLANK}}$ input on the falling edge of VCLK. It then looks at the internal RCLK signal to determine when to enable or disable SCLK at the output terminal. The decision is determined when the RCLK frequency is greater than or equal to the VCLK frequency. However, when the RCLK frequency is less than the VCLK frequency, the appearance of the SCLK waveform at the output terminal (when $\overline{\text{BLANK}}$ is sampled low on the falling edge of VCLK) can vary (see Figures B–1 and B–2).

To avoid this variation in the SCLK output waveform, the RCLK and VCLK frequencies should be chosen so that HTOTAL is evenly divisible by the ratio of the VCLK frequency to the RCLK frequency:

remainder of [HTOTAL / (VCLK frequency / RCLK frequency)] = 0

For example, if HTOTAL is even, VCLK frequency = DOTCLK frequency/8 and RCLK frequency = DOTCLK frequency/16. Then the above formula is satisfied. NOTE: When HTOTAL starts at zero, then the formula becomes:

remainder of [(HTOTAL + 1) / (VCLK frequency / SCLK frequency)] = 0.

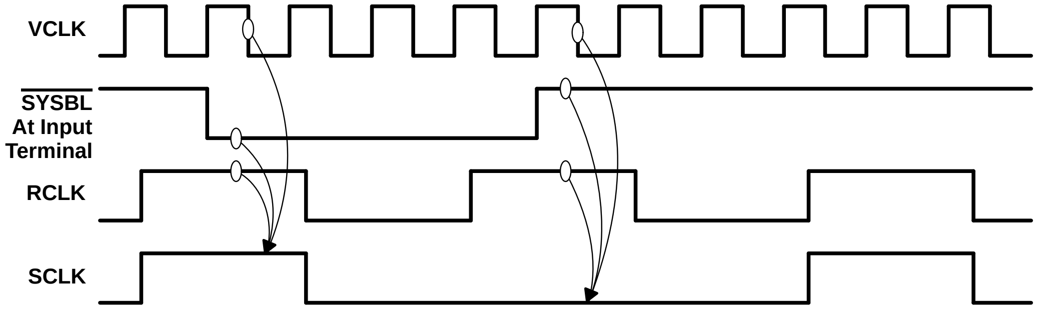


Figure B–1. VCLK and SCLK Phase Relationship (Case 1)

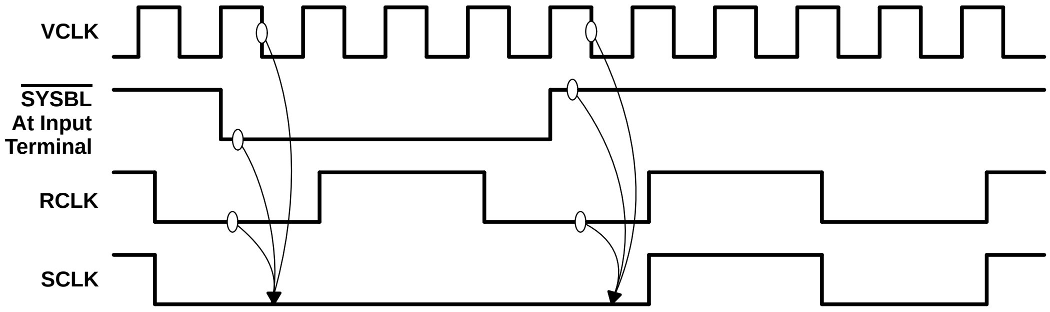


Figure B–2. VCLK and SCLK Phase Relationship (Case 2)

Appendix C
Little-Endian and Big-Endian Data Formats

It is commonly known in the computer industry that there are two different formats for memory configuration: little-endian (Intel microprocessor format) and big-endian (Motorola microprocessor-based format). When the Texas Instruments programmable pixel bus was introduced on the TLC34075 video interface palette, it allowed little-endian-based graphics-board manufacturers to design a single graphics board that could be programmed to support multiple resolutions and pixel depths. The connection of the pixel bus from the video RAM to the palette device was not a problem until big-endian-based customers desired the same capability to program their graphics designs from 1 bit/pixel (bpp), 2 bpp, 4 bpp, 8 bpp, 12 bpp, 16 bpp, 24 bpp, to 32 bpp.

For this reason, the Viewpoint video interface palette supports both little- and big-endian data formats on its pixel-bus/frame-buffer interface. The device defaults to little-endian mode at RESET (general-control register bit 3 set to logic 0) to be compatible with most PC-based systems. Big-endian-mode operation can be achieved by configuring the device to the big-endian mode (general-control register bit 3 set to logic 1) and externally reverse wiring the pixel bus from video RAM to Viewpoint on the graphics board.

The differences between the big-endian and little-endian data formats are illustrated in Figure C–1. The figure shows that the data fields representing the individual pixels in the big-endian format are in the reverse order of the little-endian format. Since the Viewpoint VIP always shifts data from low-numbered data fields to high-numbered data fields, external swapping of the pixel bus (i.e., D63 connected to P0, D0 connected to P63) ensures that the pixels are displayed on the monitor in the correct sequence. However, swapping the big-endian pixel bus causes the bits within each data field to be reversed (i.e., MSB to LSB instead of LSB to MSB). When general-control register bit 3 is set to a logic 1, unique circuitry within the TVP3020 corrects the bit sequence in each data field as it is shifted into the part. This correction is bit-plane independent and occurs regardless of whether 8, 4, 2, or 1 bits/pixel are being used.

Viewpoint also supports 12-, 16-, and 24-bit true-/direct-color for both little- and big-endian data formats on the pixel bus. By using the same wiring for big-endian operation as described above, all true-/direct-color modes are made available without hardware modification. Tables 2–8 through 2–11 give the true-/direct-color bit definitions for all modes. For example, when in one of the 16-bit true-color modes (big-endian), the first RGB data word to be displayed is located in bits 48–63 of VRAM. Swapping the external pixel bus when designing the graphics board ensures the correct display sequence by causing the first RGB word to appear at pixel bus inputs P0–P15. However, the bit order within the word is reversed. When general-control register bit 3 is set to a logic 1, the bit sequence is automatically corrected by circuitry within the TVP3020.

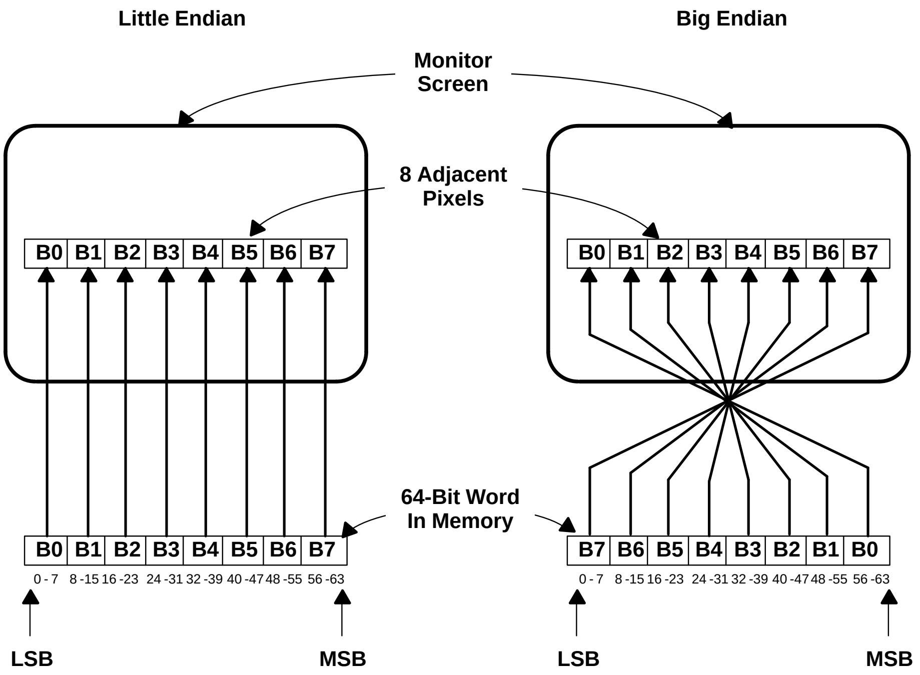


Figure C–1. Little-Endian and Big-Endian Mapping of 8-Bit/Pixel Pseudo-Color Data in Memory to Monitor Screen

Appendix D
Examples: Register Settings

Table D–1. 8-Bit/Pixel Pseudo-Color (64-Bit Pixel Bus, 8:1) Self-Clocked

REGISTER	INDEX (HEX)	SETTING (HEX)	DESCRIPTION
Input Clock Selection	1A	xx	Any clock can be chosen
Output Clock Selection	1B	5B	RCLK = SCLK = /8, VCLK = /8 (VCLK could be different)
General Control	1D	20	Default settings
Auxiliary Control	29	09	Default settings – self-clocked, palette graphics, no zoom
Color-Key Control	38	10	Default settings – pointing to palette graphics
Multiplexer-Control Register 1	18	80	Pseudo-color
Multiplexer-Control Register 2	19	1C	8 bpp, 8:1, 64-bit pixel bus width

Table D–2. 24-Bit True Color (64-Bit Pixel Bus, 2:1) Self-Clocked

REGISTER	INDEX (HEX)	SETTING (HEX)	DESCRIPTION
Input Clock Selection	1A	xx	Any clock can be chosen
Output Clock Selection	1B	49	RCLK = SCLK = /2, VCLK = /2 (VCLK could be different)
General Control	1D	20	Default settings
Auxiliary Control	29	09	Default settings – self-clocked, palette graphics, no zoom
Color-Key Control	38	10	Default settings – pointing to palette graphics
Multiplexer-Control Register 1	18	46	24-bit true color
Multiplexer-Control Register 2	19	04	24-bit true color, 64-bit pixel bus width

Table D–3. 24-Bit Direct Color (64-Bit Pixel Bus, 2:1) Self-Clocked, No Overlay

REGISTER	INDEX (HEX)	SETTING (HEX)	DESCRIPTION
Input Clock Selection	1A	xx	Any clock can be chosen
Output Clock Selection	1B	49	RCLK = SCLK = /2, VCLK = /2 (VCLK could be different)
General Control	1D	20	Default settings
Auxiliary Control	29	08	Self clocked, no window, nonpalette graphics, no zoom
Color-Key Control	38	00	Disabled – pointing to nonpalette graphics
Multiplexer-Control Register 1	18	06	24-bit direct color
Multiplexer-Control Register 2	19	1C	24-bit direct color, 64-bit pixel bus width

Table D – 4. 24-Bit Direct Color (64-Bit Pixel Bus, 2:1) Self-Clocked, Overlay PSEL Switched

REGISTER	INDEX (HEX)	SETTING (HEX)	DESCRIPTION
Input Clock Selection	1A	xx	Any clock can be chosen
Output Clock Selection	1B	49	RCLK = SCLK = /2, VCLK = /2 (VCLK could be different)
General Control	1D	20	Default settings
Auxiliary Control	29	0C	Self clocked, PSEL enabled, overlay when PSEL = 1
Color-Key Control	38	00	Disabled – pointing to nonpalette graphics
Multiplexer-Control Register 1	18	06	24-bit direct color
Multiplexer-Control Register 2	19	1C	24-bit direct color, 64-bit pixel bus width

Table D – 5. 24-Bit Direct Color (64-Bit Pixel Bus, 2:1) Self-Clocked, Overlay Auxiliary Window Switched

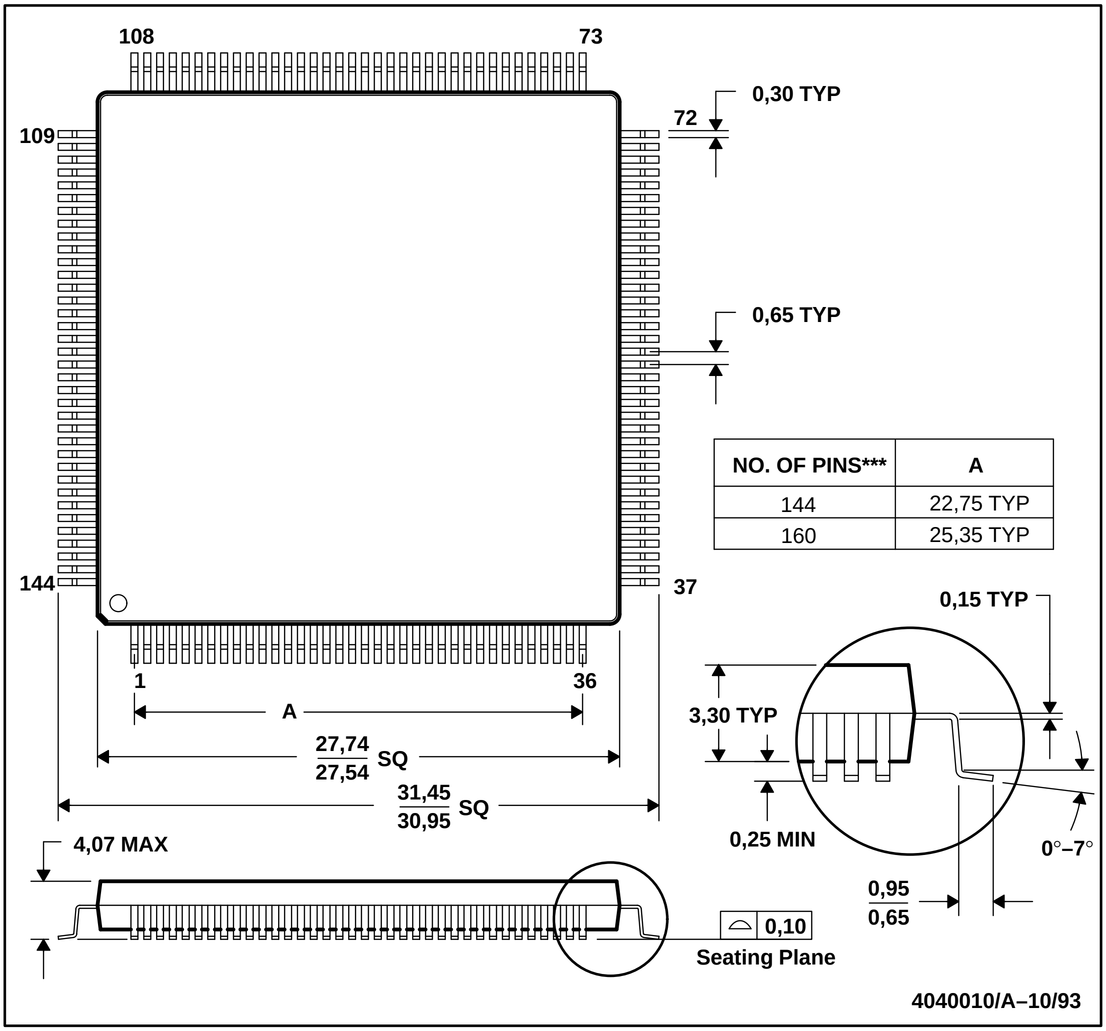
REGISTER	INDEX (HEX)	SETTING (HEX)	DESCRIPTION
Input Clock Selection	1A	xx	Any clock can be chosen
Output Clock Selection	1B	49	RCLK = SCLK = /2, VCLK = /2 (VCLK could be different)
General Control	1D	20	Default settings
Auxiliary Control	29	0A	Self clocked, auxiliary window enabled, overlay in window
Color-Key Control	38	00	Disabled – pointing to nonpalette graphics
Multiplexer-Control Register 1	18	06	24-bit direct color
Multiplexer-Control Register 2	19	1C	24-bit direct color, 64-bit pixel bus width

NOTE: For this mode, the auxiliary-window start and stop registers (Index 10 – 17) must be programmed with the appropriate window coordinates (within active display). See sections 2.16.5 and 2.16.6.

Appendix E
Mechanical Data

MDN/S-MQFP-G*** METAL QUAD (MQUAD®) CAVITY-UP FLAT PACKAGE

144-PIN SHOWN



NO. OF PINS***	A
144	22,75 TYP
160	25,35 TYP

NOTES:
A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. MQUAD is a registered trademark of Olin Corporation.
D. This quad flat package consists of a circuit mounted on a lead frame and encased within an anodized aluminum shell. The package is intended for parts requiring either a lower stress environment or higher thermal dissipation capabilities than can be supplied by plastic. Ultrasonic cleaning of this package or boards with this package is not permissible.
E. The 144 MDN is identical to 160 MDN except 4 leads per corner are removed.

Misc. Control Register Bit Definitions

General Control Register

The General Control register is used to control various functions of VIEWPOINT. It can be accessed by MPU at anytime. Bit 7 of General Control register corresponds to data bus bit 7. Index = 1D (hex).

Table J–1. General Control Register

Bit Name	Values	Description
GCR7	0: External OVS pin (default)	Overscan Control Select. Selects external pin control or internally generated overscan control. See Section 2.7.
	1: Internal	
GCR6	0: Disable (default)	Overscan Enable. Specifies whether to enable the user–defined overscan screen borders.
	1: Enable	
GCR5	0: Disable	Sync Enable. This bit specifies whether SYNC information is to be output onto IOG.
	1: Enable (default)	
GCR4	0: 0 IRE (default)	Pedestal Control. This bit specifies whether a 0 or 7.5 IRE blanking pedestal is to be generated on the video outputs. 0 IRE specifies that the black and blank levels are the same.
	1: 7.5 IRE	
GCR3	0: Little Endian (default)	Little Endian/Big Endian Select. Selects either Little or Big Endian format for the pixel bus frame buffer interface. See Sections 2.4 and 2.8.1.
	1: Big Endian	
GCR2	0: Disable (default)	Split Shift Register Transfer Enable. See Section 2.11.
	1: Enable	
GCR1	0: Active low (default)	VSYNCOUT Output Polarity. See Section 2.10.
	1: Active high	
GCR0	0: Active low (default)	HSYNCOUT Output Polarity. See Section 2.10.
	1: Active high	

Cursor Control Register

The Cursor Control register is used to control various on–chip cursor function of VIEWPOINT. It may be accessed by MPU at any time. Bit 7 of the Cursor Control register corresponds to data bus bit 7. Index = 06 (hex).

Table J–2. Cursor Control Register

Bit Name	Values	Description
CCR7	X	Reserved
	X	
CCR6	0: Disable (default)	Sprite Cursor Enable. This bit enables (logic 1) or disables (logic 0) the 64 x 64 sprite cursor.
	1: Enable	
CCR5	0: (default)	Dual Cursor Format. This bit specifies the display format at the intersection of the cross hair cursor and the user–defined cursor area. See Section 2.5.5 and the cursor intersection truth table.
	1:	
CCR4	0: XGA (default)	64 x 64 Cursor Mode Select. This bit specifies whether the XGA (logic 0) or X–windows format is used to interpret the data stored in the 64 x 64 cursor sprite RAM. See Section 2.5.2.
	1: X–Windows	
CCR3	0: Color 0 (default)	Cross Hair Color Selection. This bit specified whether the cross hair cursor is to be displayed in color 1 (logical 1) or color 0 (logical 0).
	1: Color 1	
CCR2	0: Disable (default)	Cross Hair Cursor Enable. This bit specifies whether the cross hair cursor is to be displayed (logical 1) or not (logical 0).
	1: Enable	
CCR1, CCR0	00: 1 pixel (default)	Cross Hair Thickness. These bits specify whether the vertical and horizontal thickness of the cross hair is one, three, five, or seven pixels. The segments are centered about the value in the Cursor Position (x,y) register.
	01: 3 pixels	
	10: 5 pixels	
	11: 7 pixels	

Cursor Position (x, y) Register

These registers are used to specify the (x,y) coordinate of the intersection of the cross hair cursor. They are also used in conjunction with the Sprite Origin register to specify the location of the 64 x 64 cursor area (see 2.5.3). The Cursor Position X register is made up of the Cursor Position X LSB (CPXL) and the Cursor Position X MSB (CPXM); the Cursor Position Y register is made up of the Cursor Position Y LSB (CPYL) and the Cursor Position Y MSB (CPYM). All registers are initialized to 00 (hex), and may be written to or read from by the MPU at any time. The Cursor Position is not updated until the vertical retrace interval after CPYM has been written to by the MPU.

CPXL and CPXM are cascaded to form a 12–bit Cursor Position X register. Similarly, CPYL and CPYM are cascaded to form a 12–bit Cursor Position Y register. Bits D4–D7 of CPXM and CPYM are always a logical zero.

Table J–3.

	Cursor Position X MSB								**Cursor Position X LSB**							
	(CPXM)								**(CPXL)**							
Data bit	D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
X Address	0	0	0	0	11	10	9	8	7	6	5	4	3	2	1	0
	Index = 01h								Index = 00h							

Table J–4.

	Cursor Position Y MSB								**Cursor Position Y LSB**							
	(CPYM)								**(CPYL)**							
Data bit	D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
Y Address	0	0	0	0	11	10	9	8	7	6	5	4	3	2	1	0
	Index = 03h								Index = 02h							

The Cursor Position X value to be written is calculated as follows:

Cx = desired display screen x position

Values from 0000 (hex) to 0fff (hex) may be written into the Cursor Position X register.

The Cursor Position Y value to be written is calculated as follows:

Cy = desired display screen y position

Values from 0000 (hex) to 0fff (hex) may be written into the Cursor Position Y register.

The values written into the Cursor Position X and Y registers should be relative to the first displayed pixel on the screen; i.e. (0,0).

Sprite Origin (x, y) Register

These registers are used to specify the (x,y) location of the 64 x 64 Sprite with respect to the cross hair location (see 2.5.3). The Sprite Origin X and Y registers can contain values from 0 to 63 decimal. Both registers are initialized to 1F (hex), 31 (decimal), which sets the center of the cross–hair at the center of the 64 X 64 sprite. Both registers may be written to or read from by the MPU at any time. The Sprite Origin is not updated until the vertical retrace interval after Sprite Origin X and Y registers have been written by the MPU.

Table J–5.

	Sprite Origin X							
Data bit	D7	D6	D5	D4	D3	D2	D1	D0
X Origin	0	0	Y5	4	3	2	1	0
	Index = 04h							

Table J–6.

	Sprite Origin Y							
Data bit	D7	D6	D5	D4	D3	D2	D1	D0
Y Origin	0	0	5	4	3	2	1	0
	Index = 05h							

Values from 00 (hex) to 3F (hex) may be written into the Sprite Origin X and Y registers.

Bits D6 and D7 are always 0.

Window Start (x, y) Register

These registers are used to specify the (x,y) coordinate of the upper left corner of the cross hair cursor window or auxiliary window. The Window Start X register is made up of the Window Start X LSB (WSXL) and the Window Start X MSB (WSXM); the Window Start Y register is made up of the Window Start Y LSB (WSYL) and the Window Start Y MSB (WSYM). They are not initialized and may be written to or read from by the MPU at any time. The Window Start is not updated until the vertical retrace intrval after WSYM has been written to by the MPU.

WSXL and WSXM are cascaded to form a 12–bit Window Start X register. Similarly, WSYL and WSYM are cascaded to form a 12–bit Window Start Y register. Bits D4–D7 of WSXM and WSYM are always a logical zero.

Table J–7.

	Window Start X MSB								**Window Start X LSB**							
	(WSXM)								**(WSXL)**							
Data bit	D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
X Address	0	0	0	0	11	10	9	8	7	6	5	4	3	2	1	0
	Index = 11h								Index = 10h							

Table J–8.

	Window Start Y MSB								**Window Start Y LSB**							
	(WSYM)								**(WSYL)**							
Data bit	D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
Y Address	0	0	0	0	11	10	9	8	7	6	5	4	3	2	1	0
	Index = 15h								Index = 14h							

The Window Start X value to be written is calculated as follows:

Wx = desired display screen x position

Values from 0000 (hex) to 0fff (hex) may be written into the Window Start X register.

The Window Start Y value to be written is calculated as follows:

Wy = desired display screen y position

Values from 0000 (hex) to 0fff (hex) may be written into the Window Start Y register.

The values wirtten into the Window Start X and Y registers should be relative to the first displayed pixel on the screen; i.e. (0,0).

Note that the Window Start location specified is the first location inside the window. For the cross hair cursor or auxiliary window to be displayed, the Window Start registers must specify a point on the active display.

Window Stop (x, y) Register

These registers are used to specify the (x,y) coordinate of the lower right corner of the cross hair cursor or auxiliary window. The Window Stop X register is made up of the Window Stop X LSB (WSPXL) and the Window Stop X MSB (WSPXM); the Window Stop Y register is made up of the Window Stop Y LSB (WSPYL) and the Window Stop Y MSB (WSPYM). They are not initialized and may be written to or read from by the MPU at any time. The Window Stop is not updated until the vertical retrace interval after WSPYM has been written to by the MPU.

WSPXL and WSPXM registers are cascaded to form a 12–bit Window Stop X register. Similarly, WSPYL and WSPYM registers are cascaded to form a 12–bit Window Stop Y register. Bits D4–D7 of WSPXM and WSPYM are always a logical zero.

Table J–9.

	Window Stop X MSB								**Cursor Position X LSB**							
	(WSPXM)								**(WSPXL)**							
Data bit	D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
X Address	0	0	0	0	11	10	9	8	7	6	5	4	3	2	1	0
	Index = 13h								Index = 12h							

Table J–10.

	Cursor Position X MSB								**Cursor Position X LSB**							
	(WSPYM)								**(WSPYL)**							
Data bit	D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
X Address	0	0	0	0	11	10	9	8	7	6	5	4	3	2	1	0
	Index = 17h								Index = 16h							

The Window Stop X value to be written is calculated as follows:

Wx = desired display screen x position

Values from 0000 (hex) to 0fff (hex) may be written into the Window Stop X register.

The Window Stop Y value to be written is calculated as follows:

Wy = desired display screen y position

Values from 0000 (hex) to 0fff (hex) may be written into the Window Stop Y register.

The values wirtten into the Window Stop X and Y registers should be relative to the first displayed pixel on the screen; i.e. (0,0).

Note that the Window Stop location specified is the last location inside of the window. For the cross hair cursor or auxiliary window to be displayed, the Window Start registers must specify a point on the active display.

Cursor Color 0, 1 RGB Registers

These registers are used to specify the two colors for the hardware cursor (see 2.5). They are not initialized and may be written to or read from by the MPU at any time. Note that there are 6 registers total, three for each color. The register format for both the Cursor color 0 and Cursor color 1 registers is shown below.

Table J–11.

	Cursor Color RED							
Data bit	D7	D6	D5	D4	D3	D2	D1	D0
Red Value	7	6	5	4	3	2	1	0
	Index = 23h and 26h							

Table J–12.

	Cursor Color GREEN							
Data bit	D7	D6	D5	D4	D3	D2	D1	D0
Green Value	7	6	5	4	3	2	1	0
	Index = 24h and 27h							

Table J–13.

	Cursor Color BLUE							
Data bit	D7	D6	D5	D4	D3	D2	D1	D0
Blue Value	7	6	5	4	3	2	1	0
	Index = 25h and 28h							

Values 00 (hex) to FF (hex) may be written into the Cursor Color registers.

Cursor RAM Address (LS, MS) Register

These registers are used to specify the address of where to write or read sprite cursor data from. The linear addressing scheme is depicted in Figure 4 of section 2.5.1. The Cursor RAM register is made up of the Cursor RAM LS (CRLS) and the Cursor RAM MS (CRMS). They are not initialized and may be written to by the MPU at any time.

CRLSL and CRMS are cascaded to form a 10–bit Cursor RAM Address register. Bits D2–D7 of CRMS are always a logical zero.

Table J–14.

	Cursor RAM MS Address								**Cursor RAM LS Address**							
	(CRMS)								**(CRLS)**							
Data bit	D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
Address	0	0	0	0	0	0	9	8	7	6	5	4	3	2	1	0
	Index = 09h								Index = 08h							

Index = 09 (hex)
Index = 08 (hex)

Values from 0000 (hex) to 03FF (hex) may be written into the Cursor RAM Address register.

Note that when the cursor RAM address is to be written, both registers must be written with the Cursor RAM LS Address being the first.

Cursor RAM Data Register

This register is used to read and write the contents of the sprite cursor locations whose address is specified in the Cursor RAM Address registers. The data read from or written to this register will contain 4 pixels of information, 2 bit planes per cursor pixel (see Figure 4 & section 2.5.1). The register is not initialized and may be written to or read from by the MPU at any time. The Sprite cursor data format is shown below.

Table J–15.

	Cursor RAM Data							
Data bit	D7	D6	D5	D4	D3	D2	D1	D0
Data	$P1_3$	$P0_3$	$P1_2$	$P0_2$	$P1_1$	$P0_1$	$P1_0$	$P0_0$
	Index = 0Ah							

Values from 00 (hex) to 3F (hex) may be written into the Cursor RAM Data register.

Auxiliary Control Register

The Auxiliary Control register is used to control various functions of VIEWPOINT including the Auxiliary Windowing function and Horizontal zooming. It can be accessed by MPU at anytime. Bit 7 of Auxiliary Control register corresponds to data bus bit 7. Index = 29 (hex).

Bit Name	Values	Description
ARC7	0: default	Horizontal Zoom Control. See Section 2.8.
	X	
ACR6	0: default	Horizontal Zoom Control. See Section 2.8.
	X	
ACR5	0: default	Horizontal Zoom Control. See Section 2.8.
	X	
ACR4	X	Reserved
	X	
ACR3	0: Externally clocked	Frame Buffer Clocking Select. Used to select Self or Externally Clocked timing for the Frame Buffer interface. See Section 2.3.2.
	1: Self clocked (default)	
ACR2	0: PSEL disable (default)	Windowing Function Select. Port Select Enable. See equation in Section 2.6.1.
	1: PSEL enable	
ACR1	0: Window disable (default)	Windowing Function Select. Auxiliary Window Enable. See the equation in Section 2.6.1.
	1: Window enable	
ACR0	0: True function	Windowing Function Select. Complementary Function Bit. See the equation in Section 2.6.1.
	1: Complement (default)	

Color Key Control Register

The Color Key Control register is used to control the operation of the Color Key Switching function (See section 2.6.2) It can be accessed by MPU at anytime. Bit 7 of Color Key Control register corresponds to data bus bit 7. Index = 38 (hex).

Bit Name	Values	Description
CKC7	X	Reserved
	X	
CKC6	X	Reserved
	X	
CKC5	X	Reserved
	X	
CKC4	0: True function	Color Key Function Select. Complementary function bit. See the equation in Section 2.6.2.
	1: Complement (default)	
CKC3	0: Disable compare (default)	BLUE Compare Enable. Used to enable disable the direct color BLUE field comparison. See the equation in Section 2.6.2.
	1: Enable comparison	
CKC2	0: Disable compare (default)	GREEN Compare Enable. Used to enable disable the direct color GREEN field comparison. See the equation in Section 2.6.2.
	1: Enable comparison	
CKC1	0: Disable compare (default)	RED Compare Enable. Used to enable disable the direct color RED field comparison. See the equation in Section 2.6.2.
	1: Enable comparison	
CKC0	0: Disable compare (default)	Overlay Compare Enable. Used to enable disable the direct color Overlay field comparison. See the equation in Section 2.6.2.
	1: Enable comparison	

Color Key (Red, Green, Blue, Overlay) Low and High Registers

These registers are used to specify the color comparison ranges for the four Direct Color data fields when performing color key switching. A Low and a High register are provided for each of the four data fields to facilitate the range comparison. See section 2.6 and 2.6.2 for more details on their useage. All four Low registers are initiallized with 01 (hex), while all four High registers are initiallized with FF (hex). The registers may be written to or read from by the MPU at any time. Note that there are 8 registers total, two for each color. The register format for both Low and High registers is shown below.

Table J–16.

	Color Key Low							
Data bit	D7	D6	D5	D4	D3	D2	D1	D0
Low Value	7	6	5	4	3	2	1	0
	Index = 30h, 32h, 34h, 36h							

Table J–17.

	Color Key High							
Data bit	D7	D6	D5	D4	D3	D2	D1	D0
High Value	7	6	5	4	3	2	1	0
	Index = 31h, 33h, 35h, 37h							

Values 00 (hex) to FF (hex) may be written into the 4 Color Key Low and 4 Color Key High registers.

Overscan Color RGB Registers

These registers are used to specify the color for the overscan function. This function can be used to create custom screen borders (see 2.7). They are not initialized and maybe written to or read from by the MPU at any time. The register format for both the Overscan color registers is shown below.

Table J–18.

	Overscan Color RED							
Data bit	D7	D6	D5	D4	D3	D2	D1	D0
RED Value	7	6	5	4	3	2	1	0
	Index = 20h							

Table J–19.

	Overscan Color GREEN							
Data bit	D7	D6	D5	D4	D3	D2	D1	D0
GREEN Value	7	6	5	4	3	2	1	0
	Index = 21h							

Table J–20.

	Overscan Color BLUE							
Data bit	D7	D6	D5	D4	D3	D2	D1	D0
BLUE Value	7	6	5	4	3	2	1	0
	Index = 22h							

Values 00 (hex) to FF (hex) may be written into the Overscan Color registers.

CRC LSB and MSB Registers

These registers are used to read the result of the 16–bit CRC calculation (see section 2.9). They are not initialized and may be read from by the MPU at any time. One must note, however, that they are only updated on the rising edge of the second HSYNC during vertical retrace.

CRCLSB and CRCMSB are cascaded to form a 16–bit CRC calculation remainder.

Table J–21.

	CRCMSB								**CRCLSB**							
Data bit	D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
CRC remainder	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Index = 3Dh								Index = 3Ch							

CRC Control Register

This write only register is used to specify which of the 24 DAC data lines the 16–bit CRC should be calculated on (see section 2.9). The register is not initialized and maybe written to by the MPU at any time. The CRC Control register data format is shown below.

Table J–22.

	CRC Control Register							
Data bit	D7	D6	D5	D4	D3	D2	D1	D0
Select Data	0	0	0	4	3	2	1	0
	Index = 3Eh							

Values from 00 (hex) to 23 (hex) may be written into the CRC Control register.

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.