Introduction

The RGB561 is designed to meet the large spectrum of graphics operations of high end workstations in the X Windows environment. It provides a high level of functional integration with flexibility, to achieve a variety of color modes, window attributes, monitor support, and pixel performance ranges. It supports CRT displays as well as digital video data for external devices. RGB561 can simultaneously display various visual formats with dynamic selection occurring on a pixel basis.

Microprocessor Interface (MPI)

The 8 bit microprocessor interface, $DATA_{7\text{-}0}$, is controlled by C0/C1/$\overline{CE}$/RW signals, and is used to access all configuration and control registers, window attribute tables (WATs), cursor, gamma and palette look-up tables (LUTs), MISR and DAC compare registers. It is shared with the $VIDEO_{11\text{-}0}$ bus which supports digital data output for external applications. The control signal timings are shown in "▪AC Characteristics" on page 55 and a list of the valid register addresses is summarized in Table 20 on page 27.

All registers are set to 0 by $\overline{RESET}$ with the exception of the Revision Level and reserved registers, the Color, Cursor and Gamma LUTs, the Cursor pixel map and the Window Attribute Tables, all of which must be loaded with the appropriate data through the MPI.

Programming

Address Index Register.

The RGB561 contains an internal 16 bit address register that is used as the pointer to all configuration registers, Look Up Tables (LUTs) and Window Attribute Tables (WATs). It is formed by the Address Index Low register and the Address Index High register.

The low order 8 bits of the address are contained in the Address Index Low register; they are stored by setting C1/C0 = 0/0 and writing the 8 bit value to the RGB561 Data bus. Loading the Address Index High register requires C1/C0 = 0/1 and a data write MPI cycle. To set an initial address to the register space the index register must be loaded in this ***low/high*** order to access look-up tables, WATs and the Cursor pixel map. Configuration registers do not have this low/high order requirement.

The address index register will auto increment to the next address after each read or write cycle to a configuration register or after the end of a multi-cycle LUT or WAT access. This feature can be disabled, if desired, by forcing the msb of the address index high register to '1'. The address index register will not automatically skip unused or reserved address locations in the valid address space and requires another address load or multiple accesses to move through those address locations.

Configuration Registers and Cursor pixel map.

Configuration registers, pointed to by the Address Index register, are accessed with C1/C0 = 1/0. The address index register will auto increment to the next address after each read or write cycle and can be used to sequentially initialize the control registers.

LUTs and WATs

Look up tables (Palettes and GAMMA Correction) and Window Attribute Tables, pointed to by the Address Index register, are accessed with C1/C0 = 1/1 denoting a multi-cycle access. The 10 bit ***WATs and Gamma correction tables*** are accessed from the 8 bit MPI bus in 2 cycles, and the ***3-1Kx8 Color look-up tables*** in 3 cycles, one each for red, green and blue data. If the address index register is accessed before completion of a multi-cycle access, the counter is reset and the operation is terminated.

The internal pixel clock (PIXCLK) must be active to access LUTs, WATs and cursor pixel map addresses. The PLL must be set up properly to generate the internal pixel clock or alternatively, the external pixel clock may be used. If an external pixel clock is used (EXTCLK/$\overline{EXTCLK}$), the desired frequency range (DFR) must still be programmed in the ***PLL/VCO Divider Register*** to properly access LUT RAM locations.

Reserved address space.

Writing to a reserved register address is ignored, reading from a reserved register address below X'1000 will return X'00 at the MPI. Reading reserved registers from address X'1000 to X'7FFF will tri-state the MPI $DATA_{7\text{-}0}$ outputs.

Look-Up Tables

The RGB561 has three look-up tables:

Color LUTs Consist of 3 - 1K x 8 SRAMs, one each for red, green and blue data. They are bypassed in grey scale and true color modes and are used to address the Gamma LUTs.

Gamma LUTs Consist of 3 - 256x10 SRAMs, for display color enhancement, and can also be bypassed. Their color output data drives the DACs.

Cursor LUT Contains 12 - 24 bit RGB (8/8/8) entries representing the primary and blink colors used to display the cursor and cross-hair.

Color Look-Up Table (LUT)

The three independent 1Kx8 color palettes on the RGB561 yield an indirect color look-up capability of 16.7 million colors. The palette tables can also be bypassed in any true color mode. They can be read or written through the MPI at any time. Anti-sparkle circuitry will repeat the last displayed pixel color during MPI accesses, but in some instances screen artifacts may result.

Data transfer to or from the Color LUTs is a three cycle operation with 8 bits of red, green and blue data being transferred sequentially to or from the same address before the address index register increments.

Color Lut Addressing

The Color LUT can be partitioned by the system software into LUTs for each window on the screen with a minimum color depth of 64 entries. Window partitioning and bypass are controlled by the FB_WAT and OL_WAT data bits.

In 30 bits per pixel (bpp) mode, the full 10 bit LUT address is contained in the serialized data and no starting address modification takes place. To form a 10 bit Color LUT address for all other pixel types, the appropriate pixel or overlay input data from the serializer is combined with the Color LUT starting address taken from its respective FB_WAT or OL_WAT. The process is shown in Table 1. Pixel data used is shown as the left data byte of Table 18 on page 21. The relationship between VRAM input data and the pixel or overlay data used to access the LUTs is shown in Table 7 on page 9 and Figure 7 on page 55.

Table 18 on page 21 also shows the look-up table output in bypass mode for various pixel input data for each color mode.

Table 1. Color LUT Address Generation

9	8	7	6	5	4	3	2	1	0
SA_3	SA_2	SA_1	SA_0	0	0	0	0	0	0
+		AI_7	AI_6	AI_5	AI_4	AI_3	AI_2	AI_1	AI_0
LA_9	LA_8	LA_7	LA_6	LA_5	LA_4	LA_3	LA_2	LA_1	LA_0

SA Starting address from the window attribute table
AI Address input from Table 18 on page 21.
LA Resulting color look-up table address

Gamma Correction Tables

The GAMMA look-up or correction tables are used to enhance color precision and clarity for applications requiring exact replication of color data or to compensate for display differences. The 8 bit data from the color palette is used to access the corresponding 256x10 Gamma table. In bypass mode, the 10 bit output is linearized by shifting the 8 bit address input to the output MSBs and copying the 2 input MSBs to the 2 output LSBs. Gamma tables are always bypassed when cursor pixels are displayed.

Gamma tables may also be accessed thru the MPI at any time, however, a read during active display time may result in visible screen artifacts. Loading the 10 bit color Gamma data entries requires a two cycle transfer before an address increment. Data fields on each access are shown in Table 2.

Table 2. Two cycle transfer: GAMMA entries

7	6	5	4	3	2	1	0	7	6
D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
First Access								2ND	

Cursor/Cross-Hair Look-Up Tables

The cursor and cross-hair have separate primary and blink color look-up tables each containing 3-24 bit RGB data entries, the fourth value is transparency. LUT entries are shown in Table 22 on page 45.

Cursor Pixel Map

The 64 x 64 cursor is mapped into a 1K x 8 SRAM as indicated in Tables 3 and 4. Each 8 bit input to the cursor pixel map represents 4 pixel color values. The 2 bit per pixel data is used to access the Cursor LUT to select a primary or blink color based on the value in the ***Cursor Control Register*** (X'0030), BT.

Table 3. 64x64 Cursor Pixel Screen Locations

Addr		0	1	2	3	...	60	61	62	63
X'2000	*0*	0	1	2	3	...	60	61	62	63
X'2010	*1*	64	65	66	67	...	124	125	126	127
X'2020	*2*	128	129	130	131	...	188	189	190	191
X'2030	*3*	192	193	194	195	...	252	253	254	255
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
X'23C0	*60*	3840	3841	3842	3843	...	3900	3901	3902	3903
X'23D0	*61*	3904	3905	3906	3907	...	3964	3965	3966	3967
X'23E0	*62*	3968	3969	3970	3971	...	4028	4029	4030	4031
X'23F0	*63*	4032	4033	4034	4035	...	4092	4093	4094	4095

Addr is the register address location of the first 4 pixels Register bit placement is described in Table 4.

Table 4. Cursor Pixels Mapped into 1Kx8 Memory

ADDRESS	DATA$_{7-0}$ Bits							
	7	6	5	4	3	2	1	0
X'2000	Pixel 0		Pixel 1		Pixel 2		Pixel 3	
	Bit 1	Bit 0	Bit 1	Bit 0	Bit 1	Bit 0	Bit 1	Bit 0
X'2001	Pixel 4		Pixel 5		Pixel 6		Pixel 7	
	Bit 1	Bit 0	Bit 1	Bit 0	Bit 1	Bit 0	Bit 1	Bit 0
⋮								
X'23FE	Pixel 4088		Pixel 4089		Pixel 4090		Pixel 4091	
	Bit 1	Bit 0	Bit 1	Bit 0	Bit 1	Bit 0	Bit 1	Bit 0
X'23FF	Pixel 4092		Pixel 4093		Pixel 4094		Pixel 4095	
	Bit 1	Bit 0	Bit 1	Bit 0	Bit 1	Bit 0	Bit 1	Bit 0

Window Attribute Tables (WATs)

The Window Attribute Tables (WAT) contain pixel frame buffer and overlay window display information that varies on a pixel basis. Two - 256x10 bit WATs are available, one contains the frame buffer pixel data attributes ***(FB_WAT)***, the other to store the overlay data attributes ***(OL_WAT)***.

Data for the 10 bit frame buffer and overlay window attribute tables must be loaded through the MPI in two cycles, similar to that used for the gamma correction tables. Data fields on each access are shown in Table 5.

Table 5. Two cycle transfer: WAT entries

7	6	5	4	3	2	1	0	7	6
D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
First Access								2ND	

The definition of data stored in the WAT entries is found in "Window Attribute Tables" on page 46.

Two - 16 entry 8 bit auxiliary window attribute tables ***(AUX_FB_WAT and AUX_OL_WAT)*** contain data characteristics for a group of windows. They are accessed through the MPI in a single cycle.

Segment Registers

Window ID (WID) bits are used to form the address into the WATs. When fewer than 8 bits of WID data are available for WAT access, the appropriate ***FB_WAT or OL_WAT Segment Registers*** are used to provide additional, most significant address bits.

The ***AUX_FB_WAT and AUX_OL_WAT Segment Registers*** perform the same function and augment the number of WID bits used to access the ***AUX_FB_WAT and AUX_OL_WAT*** when less than 4 bits are available . Only the least significant 4 bits of these registers are used to generate WAT addresses. The segment registers are loaded from the MPI in one cycle and are located from address X'0006-0009.

Refer to "WAT Addressing" on page 15 for a detailed description of WAT addressing and Figure 1 on page 16 for a block diagram of the registers associated with the window attribute tables.

Video

The 12 bit digital $VIDEO_{11-0}$ output of the RGB561, is used to transfer pixel data from the DAC inputs for external use (i.e. NTSC or LCD display subsystems).

The data clocked out of the VIDEO port represents the 8 MSB DAC inputs for each color. The bus is synchronized to either EXTCLK or DDOTCLK with the timings shown in Figure 8 on page 56. The 24 bit pixel DAC data is 'sliced' to provide 12 bit bus transfers. The 4 most significant bits of RGB data are provided on the rising pixel clock edge (EXTCLK or DDOTCLK) and the 4 least significant bits on the falling edge (Table 6).

The video options are controlled by bits in the ***CONF/3*** register. When video mode is enabled (VID), RGB data is output on the VIDEO bus during periods when blanking is inactive.

Since Video output bits 0-7 make use of the MPI data bus, it is imperative that all devices sharing the MPI data bus be tristated during active video. Active video time can be determined by checking the $\overline{\text{CBLANKOUT}}$ output of the RGB561 The MPI is available for register access only during blanking periods when the video mode is enabled.

Table 6. Video Output Clocking

REF/EXTCLK	11	10	9	8	7	6	5	4	3	2	1	0
RISING	RED_{7-4}				$GREEN_{7-4}$				$BLUE_{7-4}$			
FALLING	RED_{3-0}				$GREEN_{3-0}$				$BLUE_{3-0}$			

Because some monitors may be unable to sync in video mode, the DACs may be set to blank level by the Screen Control bit (CONF/2) or powered off completely using the ***DAC Control Register***, DAC bit (X'005F).

Interlace support

The RGB561 supports interlaced operation. If interlaced operation is enabled, interleave and cursor data are controlled by the ***FIELD*** input. The FIELD polarity is also controlled by the CONF/3 register.

The **FIELD** signal is used in video mode to specify whether an even (0) or odd (1) scan line is displayed. For proper interleave and cursor data generation, pixel data must be presented to the RGB561 consistent with the interlace requirements.

Frame Buffer Interface

The 200 bit VRAM input data can be configured into any of the operating modes listed below, using the serializer configuration registers. The modes determine the number of bits (pixel size) allocated for use by frame buffer and overlay layers, and the maximum number of WID bit planes configurable.

MODE	MUX Ratio	FB/OL Size	Max WID
Basic	5:1	32 bits	8 bits
Basic	4:1	32 bits	8 bits
Extended	4:1	40 bits	8 bits
Super Extended	4:1	48 bits	2 bits
30 bpp	5:1	30/8 bits	2 bits
Extended-B	8:1	20 bits	4 bits
30 bpp	4:1	30/16 bits	2 bits
Extended-A	8:1	24 bits	0 bits

The ***OVLY*** field (CONF/1) specifies the number of FB/OL bits allocated for overlay; the remaining bits are allocated to the frame buffer. Pixel format and buffer select options, specified in WAT, should not exceed the pixel size boundary. Alternatively, the boundary between frame buffer and overlay can vary on a per pixel basis, with the size of frame buffer and overlay pixels determined by the WAT. Care should be taken not to overlap the frame buffer and overlay for the given mode of operation.

Window ID bits determine the number of WID planes configured. It should not exceed the limit set by the serializer configuration. The selected number of LSB will address the WAT, the remaining bits are ignored.

When loading the configuration registers it is important to specify a valid set of parameters consistent with those available or results will be unpredictable.

The serializer configuration summary and VRAM data allocations are shown in Table 7 on page 9.

VRAM Interface

Table 7. RGB561 VRAM to PIXEL CONFIGURATION SUMMARY

VRAM DATA	BASIC		EXTENDED	SUPER_EXT	RGB 30 bpp		BASIC
	5:1 MUX	4:1 MUX	4:1 MUX	4:1 MUX	5:1 MUX	4:1 MUX	8:1 MUX
0-3	PA_{0-3}	PA_{0-3}	PA_{0-3}	PA_{0-3}	PA_{0-3}	PA_{0-3}	PA_{0-3}
4-7	PA_{4-7}	PA_{4-7}	PA_{4-7}	PA_{4-7}	PA_{4-7}	PA_{4-7}	PA_{4-7}
8-11	PA_{8-11}	PA_{8-11}	PA_{8-11}	PA_{8-11}	PA_{8-11}	PA_{8-11}	PA_{8-11}
12-15	PA_{12-15}	PA_{12-15}	PA_{12-15}	PA_{12-15}	PA_{12-15}	PA_{12-15}	PA_{12-15}
16-19	PA_{16-19}	PA_{16-19}	PA_{16-19}	PA_{16-19}	PA_{16-19}	PA_{16-19}	PB_{0-3}
20-23	PA_{20-23}	PA_{20-23}	PA_{20-23}	PA_{20-23}	PA_{20-23}	PA_{20-23}	PB_{4-7}
24-27	PB_{0-3}	PB_{0-3}	PB_{0-3}	PB_{0-3}	PB_{0-3}	PB_{0-3}	PC_{0-3}
28-31	PB_{4-7}	PB_{4-7}	PB_{4-7}	PB_{4-7}	PB_{4-7}	PB_{4-7}	PC_{4-7}
32-35	PB_{8-11}	PB_{8-11}	PB_{8-11}	PB_{8-11}	PB_{8-11}	PB_{8-11}	PC_{8-11}
36-39	PB_{12-15}	PB_{12-15}	PB_{12-15}	PB_{12-15}	PB_{12-15}	PB_{12-15}	PC_{12-15}
40-43	PB_{16-19}	PB_{16-19}	PB_{16-19}	PB_{16-19}	PB_{16-19}	PB_{16-19}	PD_{0-3}
44-47	PB_{20-23}	PB_{20-23}	PB_{20-23}	PB_{20-23}	PB_{20-23}	PB_{20-23}	PD_{4-7}
48-51	PC_{0-3}	PC_{0-3}	PC_{0-3}	PC_{0-3}	PC_{0-3}	PC_{0-3}	PE_{0-3}
52-55	PC_{4-7}	PC_{4-7}	PC_{4-7}	PC_{4-7}	PC_{4-7}	PC_{4-7}	PE_{4-7}
56-59	PC_{8-11}	PC_{8-11}	PC_{8-11}	PC_{8-11}	PC_{8-11}	PC_{8-11}	PE_{8-11}
60-63	PC_{12-15}	PC_{12-15}	PC_{12-15}	PC_{12-15}	PC_{12-15}	PC_{12-15}	PE_{12-15}
64-67	PC_{16-19}	PC_{16-19}	PC_{16-19}	PC_{16-19}	PC_{16-19}	PC_{16-19}	PF_{0-3}
68-71	PC_{20-23}	PC_{20-23}	PC_{20-23}	PC_{20-23}	PC_{20-23}	PC_{20-23}	PF_{4-7}
72-75	PD_{0-3}	PD_{0-3}	PD_{0-3}	PD_{0-3}	PD_{0-3}	PD_{0-3}	PG_{0-3}
76-79	PD_{4-7}	PD_{4-7}	PD_{4-7}	PD_{4-7}	PD_{4-7}	PD_{4-7}	PG_{4-7}
80-83	PD_{8-11}	PD_{8-11}	PD_{8-11}	PD_{8-11}	PD_{8-11}	PD_{8-11}	PG_{8-11}
84-87	PD_{12-15}	PD_{12-15}	PD_{12-15}	PD_{12-15}	PD_{12-15}	PD_{12-15}	PG_{12-15}
88-91	PD_{16-19}	PD_{16-19}	PD_{16-19}	PD_{16-19}	PD_{16-19}	PD_{16-19}	PH_{0-3}
92-95	PD_{20-23}	PD_{20-23}	PD_{20-23}	PD_{20-23}	PD_{20-23}	PD_{20-23}	PH_{4-7}
96-99	PE_{0-3}	—	PA_{24-27}	PA_{24-27}	PE_{0-3}	PA_{24-27}	PB_{8-11}
100-103	PE_{4-7}	—	PA_{28-31}	PA_{28-31}	PE_{4-7}	PA_{28-31}	PB_{12-15}
104-107	PE_{8-11}	—	PB_{24-27}	PB_{24-27}	PE_{8-11}	PB_{24-27}	PD_{8-11}
108-111	PE_{12-15}	—	PB_{28-31}	PB_{28-31}	PE_{12-15}	PB_{28-31}	PD_{12-15}
112-115	PE_{16-19}	—	PC_{24-27}	PC_{24-27}	PE_{16-19}	PC_{24-27}	PF_{8-11}
116-119	PE_{20-23}	—	PC_{28-31}	PC_{28-31}	PE_{20-23}	PC_{28-31}	PF_{12-15}
120-123	PA_{24-27}	PA_{24-27}	PA_{32-35}	PA_{40-43}	PA_{24-27}	PA_{32-35}	PA_{16-19}
124-127	PA_{28-31}	PA_{28-31}	PA_{36-39}	PA_{44-47}	PA_{28-31}	PA_{36-39}	PA_{20-23}
128-131	PB_{24-27}	PB_{24-27}	PB_{32-35}	PB_{40-43}	PB_{24-27}	PB_{32-35}	PC_{16-19}
132-135	PB_{28-31}	PB_{28-31}	PB_{36-39}	PB_{44-47}	PB_{28-31}	PB_{36-39}	PC_{20-23}
136-139	PC_{24-27}	PC_{24-27}	PC_{32-35}	PC_{40-43}	PC_{24-27}	PC_{32-35}	PE_{16-19}
140-143	PC_{28-31}	PC_{28-31}	PC_{36-39}	PC_{44-47}	PC_{28-31}	PC_{36-39}	PE_{20-23}
144-147	PD_{24-27}	PD_{24-27}	PD_{32-35}	PD_{40-43}	PD_{24-27}	PD_{32-35}	PG_{16-19}
148-151	PD_{28-31}	PD_{28-31}	PD_{36-39}	PD_{44-47}	PD_{28-31}	PD_{36-39}	PG_{20-23}
152-155	PE_{24-27}	—	PD_{24-27}	PD_{24-27}	PE_{24-27}	PD_{24-27}	PH_{8-11}
156-159	PE_{28-31}	—	PD_{28-31}	PD_{28-31}	PE_{28-31}	PD_{28-31}	PH_{12-15}
160-163	WA_{0-3}	WA_{0-3}	WA_{0-3}	WA_{0-1}/PA_{34-35}	WA_{0-1}/PA_{32-33}	WA_{0-1}/PA_{40-41}	PB_{16-19}
164-167	WA_{4-7}	WA_{4-7}	WA_{4-7}	PA_{36-39}	PA_{34-37}	PA_{42-45}	PB_{20-23}
168-171	WB_{0-3}	WB_{0-3}	WB_{0-3}	WB_{0-1}/PB_{34-35}	WB_{0-1}/PB_{32-33}	WB_{0-1}/PB_{40-41}	PD_{16-19}
172-175	WB_{4-7}	WB_{4-7}	WB_{4-7}	PB_{36-39}	PB_{34-37}	PB_{42-45}	PD_{20-23}
176-179	WC_{0-3}	WC_{0-3}	WC_{0-3}	WC_{0-1}/PC_{34-35}	WC_{0-1}/PC_{32-33}	WC_{0-1}/PC_{40-41}	PF_{16-19}
180-183	WC_{4-7}	WC_{4-7}	WC_{4-7}	PC_{36-39}	PC_{34-37}	PC_{42-45}	PF_{20-23}
184-187	WD_{0-3}	WD_{0-3}	WD_{0-3}	WD_{0-1}/PD_{34-35}	WD_{0-1}/PD_{32-33}	WD_{0-1}/PD_{40-41}	PH_{16-19}
188-191	WD_{4-7}	WD_{4-7}	WD_{4-7}	PD_{36-39}	PD_{34-37}	PD_{42-45}	PH_{20-23}
192-195	WE_{0-3}	—	—	PA_{32-33}/PB_{32-33}	WE_{0-1}/PE_{32-33}	—	$SWID_{0-3}$
196-199	WE_{4-7}	—	—	PC_{32-33}/PD_{32-33}	PE_{34-37}	—	$SWID_{4-7}$
INDEPENDENT INTERLEAVE CONFIGURATIONS				COMMON INTERLEAVE CONFIGURATIONS			

PA_{0-3} = pixel A, bits 0-3 • WA = window ID for pixel A • SWID = static WID for 8:1 mode

Table 8. RGB561 PIXEL TO VRAM CONFIGURATION SUMMARY

PIXEL DATA	BASIC		EXTENDED	SUPER_EXT	RGB 30 bpp		BASIC
	5:1 MUX	4:1 MUX	4:1 MUX	4:1 MUX	5:1 MUX	4:1 MUX	8:1 MUX
PA_{0-15}	0-15	0-15	0-15	0-15	0-15	0-15	0-15
PA_{16-23}	16-23	16-23	16-23	16-23	16-23	16-23	120-127
PA_{24-31}	120-127	120-127	96-103	96-103	120-127	96-103	—
PA_{32-35}	—	—	120-123	192,193,162,163	162-165	120-123	—
PA_{36-39}	—	—	124-127	164-167	166,167,-,-	124-127	—
PA_{40-47}	—	—	—	120-127	—	162-167,-,-	—
PB_{0-7}	24-31	24-31	24-31	24-31	24-31	24-31	16-23
PB_{8-15}	32-39	32-39	32-39	32-39	32-39	32-39	96-103
PB_{16-23}	40-47	40-47	40-47	40-47	40-47	40-47	160-167
PB_{24-31}	128-135	128-135	104-111	104-111	128-135	104-111	—
PB_{32-35}	—	—	128-131	194,195,170,171	170-173	128-131	—
PB_{36-39}	—	—	132-135	172-175	174,175,-,-	132-135	—
PB_{40-47}	—	—	—	128-135	—	170-175,-,-	—
PC_{0-15}	48-63	48-63	48-63	48-63	48-63	48-63	24-39
PC_{16-23}	64-71	64-71	64-71	64-71	64-71	64-71	128-135
PC_{24-31}	136-143	136-143	112-119	112-119	136-143	112-119	—
PC_{32-35}	—	—	136-139	196,197,178,179	178-181	136-139	—
PC_{36-39}	—	—	140-143	180-183	182,183,-,-	140-143	—
PC_{40-47}	—	—	—	136-143	—	178-183,-,-	—
PD_{0-7}	72-79	72-79	72-79	72-79	72-79	72-79	40-47
PD_{8-15}	80-87	80-87	80-87	80-87	80-87	80-87	104-111
PD_{16-23}	88-95	88-95	88-95	88-95	88-95	88-95	168-175
PD_{24-31}	144-151	144-151	152-159	152-159	144-151	152-159	—
PD_{32-35}	—	—	144-147	198,199,186,187	186-189	144-147	—
PD_{36-39}	—	—	148-151	188-191	190,191,-,-	148-151	—
PD_{40-47}	—	—	—	144-151	—	186-191,-,-	—
PE_{0-15}	96-111	—	—	—	96-111	—	48-63
PE_{16-23}	112-119	—	—	—	112-119	—	136-143
PE_{24-31}	152-159	—	—	—	152-159	—	—
PE_{32-37}	—	—	—	—	194-199	—	—
PF_{0-7}	—	—	—	—	—	—	64-71
PF_{8-15}	—	—	—	—	—	—	112-119
PF_{16-23}	—	—	—	—	—	—	176-183
PG_{0-15}	—	—	—	—	—	—	72-87
PG_{16-23}	—	—	—	—	—	—	144-151
PH_{0-7}	—	—	—	—	—	—	88-95
PH_{8-15}	—	—	—	—	—	—	152-159
PH_{16-23}	—	—	—	—	—	—	184-191
WA_{0-3}	160-163	160-163	160-163	160,161,-,-	160,161,-,-	160,161,-,-	—
WA_{4-7}	164-167	164-167	164-167	—	—	—	—
WB_{0-3}	168-171	168-171	168-171	168,169,-,-	168,169,-,-	168,169,-,-	—
WB_{4-7}	172-175	172-175	172-175	—	—	—	—
WC_{0-3}	176-179	176-179	176-179	176,177,-,-	176,177,-,-	176,177,-,-	—
WC_{4-7}	180-183	180-183	180-183	—	—	—	—
WD_{0-3}	184-187	184-187	184-187	184,185,-,-	184,185,-,-	184,185,-,-	—
WD_{4-7}	188-191	188-191	188-191	—	—	—	—
WE_{0-3}	192-195	—	—	—	192,193,-,-	—	—
WE_{4-7}	196-199	—	—	—	—	—	—
$SWID_{0-7}$	—	—	—	—	—	—	192-199
INDEPENDENT INTERLEAVE CONFIGURATIONS				COMMON INTERLEAVE CONFIGURATIONS			

PA_{0-15} = pixel A, bits 0-15 • WA = window ID for pixel A • SWID = static WID for 8:1 mode

Pixel Interpretation and Bit Assignment

<table>
<tr><th colspan="41">Table 9. RGB561 VRAM PIXEL BIT ASSIGNMENT 5:1 / 4:1 MUX Basic Mode</th></tr>
<tr><th></th><th colspan="40">BIT ASSIGNMENT</th></tr>
<tr><th></th><th>W07</th><th>W06</th><th>W05</th><th>W04</th><th>W03</th><th>W02</th><th>W01</th><th>W00</th><th>P31</th><th>P30</th><th>P29</th><th>P28</th><th>P27</th><th>P26</th><th>P25</th><th>P24</th><th>P23</th><th>P22</th><th>P21</th><th>P20</th><th>P19</th><th>P18</th><th>P17</th><th>P16</th><th>P15</th><th>P14</th><th>P13</th><th>P12</th><th>P11</th><th>P10</th><th>P09</th><th>P08</th><th>P07</th><th>P06</th><th>P05</th><th>P04</th><th>P03</th><th>P02</th><th>P01</th><th>P00</th></tr>
<tr><td>FB WAT Pixel Format</td><td colspan="40"></td></tr>
<tr><td>8 bit RGB or INDEX</td><td colspan="24"></td><td colspan="8">8_B</td><td colspan="8">8_A</td></tr>
<tr><td>12 bit RGB</td><td colspan="16"></td><td colspan="12">12_B</td><td colspan="12">12_A</td></tr>
<tr><td>16 bit RGB</td><td colspan="8"></td><td colspan="16">16_B</td><td colspan="16">16_A</td></tr>
<tr><td>24 bit RGB</td><td colspan="16"></td><td colspan="24">24</td></tr>
<tr><td>OL WAT Pixel Format</td><td colspan="40"></td></tr>
<tr><td>8 bit RGB or Index</td><td colspan="8"></td><td colspan="8">O_A</td><td colspan="8">O_B</td><td colspan="16"></td></tr>
<tr><td>6/2 bit OL/UL Index</td><td colspan="8"></td><td colspan="6">O_A</td><td colspan="2">U_A</td><td colspan="6">O_B</td><td colspan="2">U_B</td><td colspan="16"></td></tr>
<tr><td>4/4 bit OL/UL Index</td><td colspan="8"></td><td colspan="4">O_A</td><td colspan="4">U_A</td><td colspan="4">O_B</td><td colspan="4">U_B</td><td colspan="16"></td></tr>
<tr><td>4 bit OL Index</td><td colspan="8"></td><td colspan="4">O_B</td><td colspan="4">O_A</td><td colspan="24"></td></tr>
<tr><td>WID bits</td><td colspan="40"></td></tr>
<tr><td>8 bit WID</td><td colspan="8">W</td><td colspan="32"></td></tr>
<tr><td colspan="41">Note: Subscripts A and B refer to Buffer A and Buffer B respectively for double buffer modes.</td></tr>
</table>

<table>
<tr><th colspan="49">Table 10. RGB561 VRAM PIXEL BIT ASSIGNMENT 4:1 MUX Extended Mode</th></tr>
<tr><th></th><th colspan="48">BIT ASSIGNMENT</th></tr>
<tr><th></th><th>W07</th><th>W06</th><th>W05</th><th>W04</th><th>W03</th><th>W02</th><th>W01</th><th>W00</th><th>P39</th><th>P38</th><th>P37</th><th>P36</th><th>P35</th><th>P34</th><th>P33</th><th>P32</th><th>P31</th><th>P30</th><th>P29</th><th>P28</th><th>P27</th><th>P26</th><th>P25</th><th>P24</th><th>P23</th><th>P22</th><th>P21</th><th>P20</th><th>P19</th><th>P18</th><th>P17</th><th>P16</th><th>P15</th><th>P14</th><th>P13</th><th>P12</th><th>P11</th><th>P10</th><th>P09</th><th>P08</th><th>P07</th><th>P06</th><th>P05</th><th>P04</th><th>P03</th><th>P02</th><th>P01</th><th>P00</th></tr>
<tr><td>FB WAT Pixel Format</td><td colspan="48"></td></tr>
<tr><td>8 bit RGB or INDEX</td><td colspan="32"></td><td colspan="8">8_B</td><td colspan="8">8_A</td></tr>
<tr><td>12 bit RGB</td><td colspan="24"></td><td colspan="12">12_B</td><td colspan="12">12_A</td></tr>
<tr><td>16 bit RGB</td><td colspan="16"></td><td colspan="16">16_B</td><td colspan="16">16_A</td></tr>
<tr><td>24 bit RGB</td><td colspan="24"></td><td colspan="24">24</td></tr>
<tr><td>OL WAT Pixel Format</td><td colspan="48"></td></tr>
<tr><td>8 bit RGB or Index</td><td colspan="8"></td><td colspan="8">O_A</td><td colspan="8">O_B</td><td colspan="24"></td></tr>
<tr><td>6/2 bit OL/UL Index</td><td colspan="8"></td><td colspan="6">O_A</td><td colspan="2">U_A</td><td colspan="6">O_B</td><td colspan="2">U_B</td><td colspan="24"></td></tr>
<tr><td>4/4 bit OL/UL Index</td><td colspan="8"></td><td colspan="4">O_A</td><td colspan="4">U_A</td><td colspan="4">O_B</td><td colspan="4">U_B</td><td colspan="24"></td></tr>
<tr><td>4 bit OL Index</td><td colspan="8"></td><td colspan="4">O_B</td><td colspan="4">O_A</td><td colspan="32"></td></tr>
<tr><td>WID bits</td><td colspan="48"></td></tr>
<tr><td>8 bit WID</td><td colspan="8">W</td><td colspan="40"></td></tr>
<tr><td colspan="49">Note: Subscripts A and B refer to Buffer A and Buffer B respectively for double buffer modes.</td></tr>
</table>

<table>
<tr><th colspan="51">Table 11. RGB561 VRAM PIXEL BIT ASSIGNMENT 4:1 MUX Super Extended mode</th></tr>
<tr><th></th><th colspan="50">BIT ASSIGNMENT</th></tr>
<tr><th></th><th>W01</th><th>W00</th><th>P47</th><th>P46</th><th>P45</th><th>P44</th><th>P43</th><th>P42</th><th>P41</th><th>P40</th><th>P39</th><th>P38</th><th>P37</th><th>P36</th><th>P35</th><th>P34</th><th>P33</th><th>P32</th><th>P31</th><th>P30</th><th>P29</th><th>P28</th><th>P27</th><th>P26</th><th>P25</th><th>P24</th><th>P23</th><th>P22</th><th>P21</th><th>P20</th><th>P19</th><th>P18</th><th>P17</th><th>P16</th><th>P15</th><th>P14</th><th>P13</th><th>P12</th><th>P11</th><th>P10</th><th>P09</th><th>P08</th><th>P07</th><th>P06</th><th>P05</th><th>P04</th><th>P03</th><th>P02</th><th>P01</th><th>P00</th></tr>
<tr><td>FB WAT Pixel Format</td><td colspan="50"></td></tr>
<tr><td>8 bit RGB or INDEX</td><td colspan="34"></td><td colspan="8">8_B</td><td colspan="8">8_A</td></tr>
<tr><td>12 bit RGB</td><td colspan="26"></td><td colspan="12">12_B</td><td colspan="12">12_A</td></tr>
<tr><td>16 bit RGB</td><td colspan="18"></td><td colspan="16">16_B</td><td colspan="16">16_A</td></tr>
<tr><td>24 bit RGB</td><td colspan="2"></td><td colspan="24">24_B</td><td colspan="24">24_A</td></tr>
<tr><td>OL WAT Pixel Format</td><td colspan="50"></td></tr>
<tr><td>8 bit RGB or Index</td><td colspan="2"></td><td colspan="8">O_A</td><td colspan="8">O_B</td><td colspan="32"></td></tr>
<tr><td>6/2 bit OL/UL Index</td><td colspan="2"></td><td colspan="6">O_A</td><td colspan="2">U_A</td><td colspan="6">O_B</td><td colspan="2">U_B</td><td colspan="32"></td></tr>
<tr><td>4/4 bit OL/UL Index</td><td colspan="2"></td><td colspan="4">O_A</td><td colspan="4">U_A</td><td colspan="4">O_B</td><td colspan="4">U_B</td><td colspan="32"></td></tr>
<tr><td>4 bit OL Index</td><td colspan="2"></td><td colspan="4">O_B</td><td colspan="4">O_A</td><td colspan="40"></td></tr>
<tr><td>WID bits</td><td colspan="50"></td></tr>
<tr><td>2 bit WID</td><td colspan="2">W</td><td colspan="48"></td></tr>
<tr><td colspan="51">Note: Subscripts A and B refer to Buffer A and Buffer B respectively for double buffer modes.</td></tr>
</table>

<table>
<tr><td colspan="41">Table 12. RGB561 VRAM PIXEL BIT ASSIGNMENT 5:1 MUX, 30 bit/pixel Mode</td></tr>
<tr><td></td><td colspan="40">BIT ASSIGNMENT</td></tr>
<tr><td></td><td>W01</td><td>W00</td><td>P37</td><td>P36</td><td>P35</td><td>P34</td><td>P33</td><td>P32</td><td>P31</td><td>P30</td><td>P29</td><td>P28</td><td>P27</td><td>P26</td><td>P25</td><td>P24</td><td>P23</td><td>P22</td><td>P21</td><td>P20</td><td>P19</td><td>P18</td><td>P17</td><td>P16</td><td>P15</td><td>P14</td><td>P13</td><td>P12</td><td>P11</td><td>P10</td><td>P09</td><td>P08</td><td>P07</td><td>P06</td><td>P05</td><td>P04</td><td>P03</td><td>P02</td><td>P01</td><td>P00</td></tr>
<tr><td>FB WAT Pixel Format</td><td colspan="40"></td></tr>
<tr><td>8 bit RGB or INDEX</td><td colspan="24"></td><td colspan="8">8_B</td><td colspan="8">8_A</td></tr>
<tr><td>12 bit RGB</td><td colspan="16"></td><td colspan="12">12_B</td><td colspan="12">12_A</td></tr>
<tr><td>16 bit RGB</td><td colspan="8"></td><td colspan="16">16_B</td><td colspan="16">16_A</td></tr>
<tr><td>30 bit RGB (cf=RGB)</td><td colspan="2"></td><td colspan="6">BBGGRR</td><td colspan="8"></td><td colspan="8">B</td><td colspan="8">G</td><td colspan="8">R</td></tr>
<tr><td>30 bit RGB (cf=BGR)</td><td colspan="2"></td><td colspan="6">BBGGRR</td><td colspan="8"></td><td colspan="8">R</td><td colspan="8">G</td><td colspan="8">B</td></tr>
<tr><td>OL WAT Pixel Format</td><td colspan="40"></td></tr>
<tr><td>8 bit RGB or Index</td><td colspan="8"></td><td colspan="8">O_A</td><td colspan="8">O_B</td><td colspan="16"></td></tr>
<tr><td>6/2 bit OL/UL Index</td><td colspan="8"></td><td colspan="6">O_A</td><td colspan="2">U_A</td><td colspan="6">O_B</td><td colspan="2">U_B</td><td colspan="16"></td></tr>
<tr><td>4/4 bit OL/UL Index</td><td colspan="8"></td><td colspan="4">O_A</td><td colspan="4">U_A</td><td colspan="4">O_B</td><td colspan="4">U_B</td><td colspan="16"></td></tr>
<tr><td>4 bit OL Index</td><td colspan="8"></td><td colspan="4">O_B</td><td colspan="4">O_A</td><td colspan="24"></td></tr>
<tr><td>WID bits</td><td colspan="40"></td></tr>
<tr><td>2 bit WID</td><td colspan="2">W</td><td colspan="38"></td></tr>
<tr><td colspan="41">Note: Subscripts A and B refer to Buffer A and Buffer B respectively for double buffer modes.</td></tr>
</table>

<table>
<tr><td colspan="49">Table 13. RGB561 VRAM PIXEL BIT ASSIGNMENT 4:1 MUX 30 bit/pixel Mode</td></tr>
<tr><td></td><td colspan="48">BIT ASSIGNMENT</td></tr>
<tr><td></td><td>W01</td><td>W00</td><td>P45</td><td>P44</td><td>P43</td><td>P42</td><td>P41</td><td>P40</td><td>P39</td><td>P38</td><td>P37</td><td>P36</td><td>P35</td><td>P34</td><td>P33</td><td>P32</td><td>P31</td><td>P30</td><td>P29</td><td>P28</td><td>P27</td><td>P26</td><td>P25</td><td>P24</td><td>P23</td><td>P22</td><td>P21</td><td>P20</td><td>P19</td><td>P18</td><td>P17</td><td>P16</td><td>P15</td><td>P14</td><td>P13</td><td>P12</td><td>P11</td><td>P10</td><td>P09</td><td>P08</td><td>P07</td><td>P06</td><td>P05</td><td>P04</td><td>P03</td><td>P02</td><td>P01</td><td>P00</td></tr>
<tr><td>FB WAT Pixel Format</td><td colspan="48"></td></tr>
<tr><td>8 bit RGB or INDEX</td><td colspan="32"></td><td colspan="8">8_B</td><td colspan="8">8_A</td></tr>
<tr><td>12 bit RGB</td><td colspan="24"></td><td colspan="12">12_B</td><td colspan="12">12_A</td></tr>
<tr><td>16 bit RGB</td><td colspan="16"></td><td colspan="16">16_B</td><td colspan="16">16_A</td></tr>
<tr><td>30 bit RGB (cf=RGB)</td><td colspan="2"></td><td colspan="6">BBGGRR</td><td colspan="16"></td><td colspan="8">B</td><td colspan="8">G</td><td colspan="8">R</td></tr>
<tr><td>30 bit RGB (cf=BGR)</td><td colspan="2"></td><td colspan="6">BBGGRR</td><td colspan="16"></td><td colspan="8">R</td><td colspan="8">G</td><td colspan="8">B</td></tr>
<tr><td>OL WAT Pixel Format</td><td colspan="48"></td></tr>
<tr><td>8 bit RGB or Index</td><td colspan="8"></td><td colspan="8">O_A</td><td colspan="8">O_B</td><td colspan="24"></td></tr>
<tr><td>6/2 bit OL/UL Index</td><td colspan="8"></td><td colspan="6">O_A</td><td colspan="2">U_A</td><td colspan="6">O_B</td><td colspan="2">U_B</td><td colspan="24"></td></tr>
<tr><td>4/4 bit OL/UL Index</td><td colspan="8"></td><td colspan="4">O_A</td><td colspan="4">U_A</td><td colspan="4">O_B</td><td colspan="4">U_B</td><td colspan="24"></td></tr>
<tr><td>4 bit OL Index</td><td colspan="8"></td><td colspan="4">O_B</td><td colspan="4">O_A</td><td colspan="32"></td></tr>
<tr><td>WID bits</td><td colspan="48"></td></tr>
<tr><td>2 bit WID</td><td colspan="2">W</td><td colspan="46"></td></tr>
<tr><td colspan="49">Note: Subscripts A and B refer to Buffer A and Buffer B respectively for double buffer modes.</td></tr>
</table>

Table 14. RGB561 VRAM PIXEL BIT ASSIGNMENT 8:1 MUX Mode A

	BIT ASSIGNMENT																							
	P23	P22	P21	P20	P19	P18	P17	P16	P15	P14	P13	P12	P11	P10	P09	P08	P07	P06	P05	P04	P03	P02	P01	P00
FB WAT Pixel Format																								
8 bit RGB or INDEX									8_B								8_A							
12 bit RGB	12_B												12_A											
16 bit RGB									16															
24 bit RGB	24																							
OL WAT Pixel Format																								
8 bit RGB or Index	O_A								O_B															
6/2 bit OL/UL Index	O_A						U_A		O_B						U_B									
4/4 bit OL/UL Index	O_A				U_A				O_B				U_B											
4 bit OL Index	O_B				O_A																			
WID bits																								
8 bit WID	W																							

Note: Subscripts A and B refer to Buffer A and Buffer B respectively for double buffer modes.

Table 15. RGB561 VRAM PIXEL BIT ASSIGNMENT 8:1 MUX Mode B

	BIT ASSIGNMENT																							
	P23	P22	P21	P20	P19	P18	P17	P16	P15	P14	P13	P12	P11	P10	P09	P08	P07	P06	P05	P04	P03	P02	P01	P00
FB WAT Pixel Format																								
8 bit RGB or INDEX									8_B								8_A							
12 bit RGB													12											
16 bit RGB									16															
OL WAT Pixel Format																								
8 bit RGB or Index	$O_{A(3:0)}$								$O_{A(7:4)}$															
6/2 bit OL/UL Index	$O_{A(1:0)}$		$U_{A(1:0)}$						$O_{A(5:2)}$															
4/4 bit OL/UL Index	$U_{A(3:0)}$								$O_{A(3:0)}$															
4 bit OL Index	$O_{A(3:0)}$								$O_{B(3:0)}$															
WID bits																								
4 bit WID					W																			

Note: Subscripts A and B refer to Buffer A and Buffer B respectively for double buffer modes.

Pixel Interleave

Interleave is a technique in which the order of the pixels, for each scan line, are varied to enhance performance on the rendering side of frame buffer. It attempts to match the difference between the way tiled pixels are mapped to the screen and stored in the frame buffer to achieve symmetrical rendering performance for both horizontal and vertical screen updates. The letters A/B/C/D/E represent pixels and also represent the different VRAM modules that store the pixel data. All interleave modes (except 000) store consecutive pixels of a vertical or diagonal line in different VRAMs, allowing simultaneous frame buffer update by the rasterizer or controller.

The ***Interleave Control Register*** selects the interleave mode desired, the starting pixel on the scan line and whether or not OL/WID bits should be interleaved. Data presented to the RGB561 can be interleaved in any of the multiplexing modes. Four methods are available in 4:1, 6 in 5:1 and 8 in 8:1. The tables indicate the methods and results of the available interleave pattern. The Configuration Summary, Table 7, identifies the modes which allow independent OL/WID interleave selection and those which interleave OL/WID data together with the pixel pattern.

INTERLEAVE MODE	5:1 MULTIPLEXING					4:1 MULTIPLEXING			
STARTING PIXEL⇒	A	B	C	D	E	A	B	C	D
000	ABCDE	BCDEA	CDEAB	DEABC	EABCD	ABCD	BCDA	CDAB	DABC
	.	.	.	.	.	.	.	.	.
001	ABCDE	BCDEA	CDEAB	DEABC	EABCD	ABCD	BCDA	CDAB	DABC
	BCDEA	CDEAB	DEABC	EABCD	ABCDE	BCDA	CDAB	DABC	ABCD
	CDEAB	DEABC	EABCD	ABCDE	BCDEA	CDAB	DABC	ABCD	BCDA
	DEABC	EABCD	ABCDE	BCDEA	CDEAB	DABC	ABCD	BCDA	CDAB
	EABCD	ABCDE	BCDEA	CDEAB	DEABC	.	.	.	.
010	ABCDE	BCDEA	CDEAB	DEABC	EABCD	ABCD	BCDA	CDAB	DABC
	CDEAB	DEABC	EABCD	ABCDE	BCDEA	CDAB	DABC	ABCD	BCDA
	EABCD	ABCDE	BCDEA	CDEAB	DEABC	.	.	.	.
	BCDEA	CDEAB	DEABC	EABCD	ABCDE				
	DEABC	EABCD	ABCDE	BCDEA	CDEAB				
011	ABCDE	BCDEA	CDEAB	DEABC	EABCD	ABCD	BCDA	CDAB	DABC
	DEABC	EABCD	ABCDE	BCDEA	CDEAB	DABC	ABCD	BCDA	CDAB
	BCDEA	CDEAB	DEABC	EABCD	ABCDE	CDAB	DABC	ABCD	BCDA
	EABCD	ABCDE	BCDEA	CDEAB	DEABC	BCDA	CDAB	DABC	ABCD
	CDEAB	DEABC	EABCD	ABCDE	BCDEA	.	.	.	.
100	ABCDE	BCDEA	CDEAB	DEABC	EABCD				
	EABCD	ABCDE	BCDEA	CDEAB	DEABC		Not	Valid	
	DEABC	EABCD	ABCDE	BCDEA	CDEAB				
	CDEAB	DEABC	EABCD	ABCDE	BCDEA				
	BCDEA	CDEAB	DEABC	EABCD	ABCDE				
101	ABCDE	BCDEA	CDEAB		EABCD				
	CDEAB	ABCDE	EABCD	Not	BCDEA		Not	Valid	
	EABCD	CDEAB	BCDEA	Valid	ABCDE				
	BCDEA	EABCD	ABCDE		CDEAB				
110		Not		Valid			Not	Valid	
111		Not		Valid			Not	Valid	

INTERLEAVE MODE	8:1 MULTIPLEXING							
STARTING PIXEL⇒	A	B	C	D	E	F	G	H
000	ABCDEFGH	BCDEFGHA	CDEFGHAB	DEFGHABC	EFGHABCD	FGHABCDE	GHABCDEF	HABCDEFG
	.	.	.	.	.	.	.	.
001	ABCDEFGH	BCDEFGHA	CDEFGHAB	DEFGHABC	EFGHABCD	FGHABCDE	GHABCDEF	HABCDEFG
	BCDEFGHA	CDEFGHAB	DEFGHABC	EFGHABCD	FGHABCDE	GHABCDEF	HABCDEFG	ABCDEFGH
	CDEFGHAB	DEFGHABC	EFGHABCD	FGHABCDE	GHABCDEF	HABCDEFG	ABCDEFGH	BCDEFGHA
	DEFGHABC	EFGHABCD	FGHABCDE	GHABCDEF	HABCDEFG	ABCDEFGH	BCDEFGHA	CDEFGHAB
	EFGHABCD	FGHABCDE	GHABCDEF	HABCDEFG	ABCDEFGH	BCDEFGHA	CDEFGHAB	DEFGHABC
	FGHABCDE	GHABCDEF	HABCDEFG	ABCDEFGH	BCDEFGHA	CDEFGHAB	DEFGHABC	EFGHABCD
	GHABCDEF	HABCDEFG	ABCDEFGH	BCDEFGHA	CDEFGHAB	DEFGHABC	EFGHABCD	FGHABCDE
	HABCDEFG	ABCDEFGH	BCDEFGHA	CDEFGHAB	DEFGHABC	EFGHABCD	FGHABCDE	GHABCDEF
010	ABCDEFGH	BCDEFGHA	CDEFGHAB	DEFGHABC	EFGHABCD	FGHABCDF	GHABCDEF	HABCDEFG
	CDEFGHAB	DEFGHABC	EFGHABCD	FGHABCDE	GHABCDEF	HABCDEFG	ABCDEFGH	BCDEFGHA
	EFGHABCD	FGHABCDE	GHABCDEF	HABCDEFG	ABCDEFGH	BCDEFGHA	CDEFGHAB	DEFGHABC
	GHABCDEF	HABCDEFG	ABCDEFGH	BCDEFGHA	CDEFGHAB	DEFGHABC	EFGHABCD	FGHABCDE
	.	.	.	.	.	.	.	.
011	ABCDEFGH	BCDEFGHA	CDEFGHAB	DEFGHABC	EFGHABCD	FGHABCDE	GHABCDEF	HABCDEFG
	DEFGHABC	EFGHABCD	FGHABCDE	GHABCDEF	HABCDEFG	ABCDEFGH	BCDEFGHA	CDEFGHAB
	GHABCDEF	HABCDEFG	ABCDEFGH	BCDEFGHA	CDEFGHAB	DEFGHABC	EFGHABCD	FGHABCDE
	BCDEFGHA	CDEFGHAB	DEFGHABC	EFGHABCD	FGHABCDE	GHABCDEF	HABCDEFG	ABCDEFGH
	EFGHABCD	FGHABCDE	GHABCDEF	HABCDEFG	ABCDEFGH	BCDEFGHA	CDEFGHAB	DEFGHABC
	HABCDEFG	ABCDEFGH	BCDEFGHA	CDEFGHAB	DEFGHABC	EFGHABCD	FGHABCDE	GHABCDEF
	CDEFGHAB	DEFGHABC	EFGHABCD	FGHABCDE	GHABCDEF	HABCDEFG	ABCDEFGH	BCDEFGHA
	FGHABCDE	GHABCDEF	HABCDEFG	ABCDEFGH	BCDEFGHA	CDEFGHAB	DEFGHABC	EFGHABCD
100	ABCDEFGH	BCDEFGHA	CDEFGHAB	DEFGHABC	EFGHABCD	FGHABCDE	GHABCDEF	HABCDEFG
	EFGHABCD	FGHABCDE	GHABCDEF	HABCDEFG	ABCDEFGH	BCDEFGHA	CDEFGHAB	DEFGHABC
	.	.	.	.	.	.	.	.
101	ABCDEFGH	BCDEFGHA	CDEFGHAB	DEFGHABC	EFGHABCD	FGHABCDE	GHABCDEF	HABCDEFG
	FGHABCDE	GHABCDEF	HABCDEFG	ABCDEFGH	BCDEFGHA	CDEFGHAB	DEFGHABC	EFGHABCD
	CDEFGHAB	DEFGHABC	EFGHABCD	FGHABCDE	GHABCDEF	HABCDEFG	ABCDEFGH	BCDEFGHA
	HABCDEFG	ABCDEFGH	BCDEFGHA	CDEFGHAB	DEFGHABC	EFGHABCD	FGHABCDE	GHABCDEF
	EFGHABCD	FGHABCDE	GHABCDEF	HABCDEFG	ABCDEFGH	BCDEFGHA	CDEFGHAB	DEFGHABC
	BCDEFGHA	CDEFGHAB	DEFGHABC	EFGHABCD	FGHABCDE	GHABCDEF	HABCDEFG	ABCDEFGH
	GHABCDEF	HABCDEFG	ABCDEFGH	BCDEFGHA	CDEFGHAB	DEFGHABC	EFGHABCD	FGHABCDE
	DEFGHABC	EFGHABCD	FGHABCDE	GHABCDEF	HABCDEFG	ABCDEFGH	BCDEFGHA	CDEFGHAB
110	ABCDEFGH	BCDEFGHA	CDEFGHAB	DEFGHABC	EFGHABCD	FGHABCDE	GHABCDEF	HABCDEFG
	GHABCDEF	HABCDEFG	ABCDEFGH	BCDEFGHA	CDEFGHAB	DEFGHABC	EFGHABCD	FGHABCDE
	EFGHABCD	FGHABCDE	GHABCDEF	HABCDEFG	ABCDEFGH	BCDEFGHA	CDEFGHAB	DEFGHABC
	CDEFGHAB	DEFGHABC	EFGHABCD	FGHABCDE	GHABCDEF	HABCDEFG	ABCDEFGH	BCDEFGHA
	.	.	.	.	.	.	.	.
111	ABCDEFGH	BCDEFGHA	CDEFGHAB	DEFGHABC	EFGHABCD	FGHABCDE	GHABCDEF	HABCDEFG
	HABCDEFG	ABCDEFGH	BCDEFGHA	CDEFGHAB	DEFGHABC	EFGHABCD	FGHABCDE	GHABCDEF
	GHABCDEF	HABCDEFG	ABCDEFGH	BCDEFGHA	CDEFGHAB	DEFGHABC	EFGHABCD	FGHABCDE
	FGHABCDE	GHABCDEF	HABCDEFG	ABCDEFGH	BCDEFGHA	CDEFGHAB	DEFGHABC	EFGHABCD
	EFGHABCD	FGHABCDE	GHABCDEF	HABCDEFG	ABCDEFGH	BCDEFGHA	CDEFGHAB	DEFGHABC
	DEFGHABC	EFGHABCD	FGHABCDE	GHABCDEF	HABCDEFG	ABCDEFGH	BCDEFGHA	CDEFGHAB
	CDEFGHAB	DEFGHABC	EFGHABCD	FGHABCDE	GHABCDEF	HABCDEFG	ABCDEFGH	BCDEFGHA
	BCDEFGHA	CDEFGHAB	DEFGHABC	EFGHABCD	FGHABCDE	GHABCDEF	HABCDEFG	ABCDEFGH

Window Attribute Tables

Window IDs - Window Attribute Tables

The WIDs are used to access the attributes of their associated pixel (FB) and overlay (OL) data. These attributes are stored in four Window Attribute Tables (WATs):

- FB_WAT
- OL_WAT
- AUX_FB_WAT
- AUX_OL_WAT

The WATs specify window characteristics that vary on a pixel basis: color mode, Color LUT start address, frame buffer data selection criteria and pixel format, for example. The auxiliary WATs contain attributes which apply to a group of windows, such as Gamma correction or cross-hair usage.

The RGB561 supports up to 8 window ID planes. The WID bits may be used as a common address for both the FB_WAT and OL_WAT, or they may be split into separate FB and OL address bits. Any missing address bits are supplied by the appropriate segment registers, and used to access frame buffer pixel attributes in the FB_WAT and overlay pixel attributes in the OL_WAT.

When in the 8:1 mux mode and 0 WID bits are specified in the ***LWID*** field (CONF/1), the STATIC WID bits are used to access the WATs. The STATIC WID bits are NOT serialized, and therefore a fixed WAT entry will be selected based on their values. These bits are used on a 'MODE' or 'FRAME' basis, they are not latched internally and must remain stable for the entire screen or unexpected results may occur.

WAT Addressing

The FB_WAT and OL_WAT have 256 entries, the AUX_FB_WAT and AUX_OL_WAT have 16 entries. Whenever less than 8 WID bits are available for FB_WAT and OL_WAT addressing, the ***FB Segment Register and OL Segment Register*** are used to provide the additional address data. This is also true when generating the 4 bits necessary to access the AUX_FB_WAT and AUX_OL_WAT using data stored in the ***AUX_FB Segment Register and AUX_OL Segment Register***.

The STATIC WID bits replace the Segment Registers in the 8:1 mode when ***LWID*** = 0.

AUX_WAT Addressing

The AUX_WATs can be aligned in two ways to the corresponding FB or OL WAT. See Figure 4 on page 18 for the msb and lsb address align formats as controlled by CONF/4, AOW and AFW.

WAT Addressing Architecture

Figure 1 shows a block diagram of the components used for WAT addressing. Figure 2, shows formation of the FB_WAT address, with WID bits common to both frame buffer and overlay pixel data, using contents of the ***AUX_FB Segment Register and FB Segment Register*** as necessary. This formation of the OL address is identical to that shown using the ***AUX_OL Segment Register and OL Segment Register***. When all 8 WID bits are common to frame buffer and overlay data, 256 FB_WAT entries can be addressed and 16 FB_WAT entries share a single entry in the AUX_FB_WAT.

If only 4 WID bits are available for the frame buffer, each of the 16 addressable FB_WAT entries corresponds to an entry in the AUX_FB_WAT. In Figure 3 is shown address formation for each of the four WATs with 8 and 6 WID bits split between the frame buffer and overlay pixels.

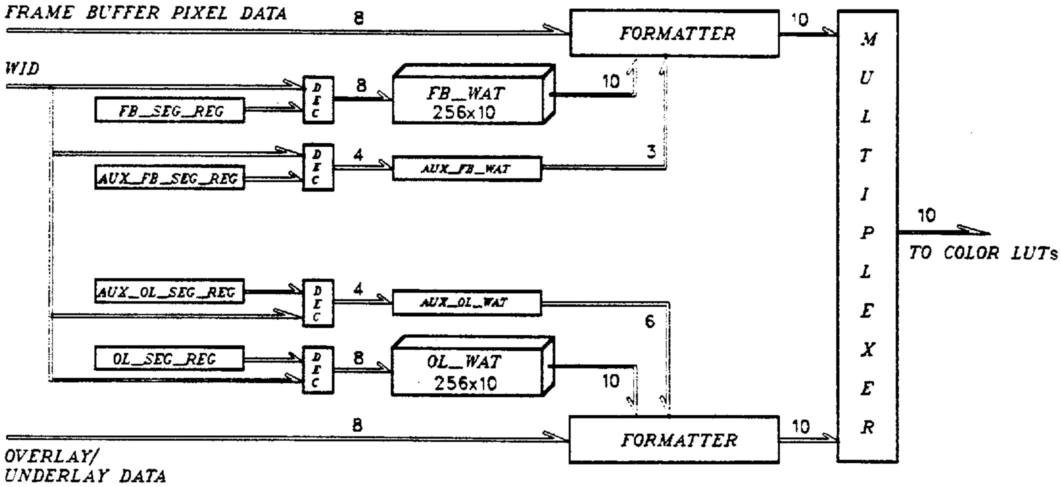


Figure 1. WAT Addressing:. Block Diagram

COMMON MODE

	7	6	5	4	3	2	1	0
MAX WID Bits	W7	W6	W5	W4	W3	W2	W1	W0
FB Segment Register	FS7	FS6	FS5	FS4	FS3	FS2	FS1	FS0

	3	2	1	0
AUX_FB Segment Reg	AF3	AF2	AF1	AF0

Logical WID Bits (CONF/:)	FB_WAT Address	AUX_FB Address CONF/4, AFW=0	AUX_FB Address CONF/4, AFW=1
0	FS7 FS6 FS5 FS4 FS3 FS2 FS1 FS0	AF3 AF2 AF1 AF0	AF3 AF2 AF1 AF0
2	FS7 FS6 FS5 FS4 FS3 FS2 W1 W0	AF3 AF2 W1 W0	AF3 AF2 W1 W0
4	FS7 FS6 FS5 FS4 W3 W2 W1 W0	W3 W2 W1 W0	W3 W2 W1 W0
6	FS7 FS6 W5 W4 W3 W2 W1 W0	W3 W2 W1 W0	W5 W4 W3 W2
8	W7 W6 W5 W4 W3 W2 W1 W0	W3 W2 W1 W0	W7 W6 W5 W4

Figure 2. WAT Addressing: Common WID Mode. AUX_OL_WAT Addressing uses AUX_OL Segment Register data

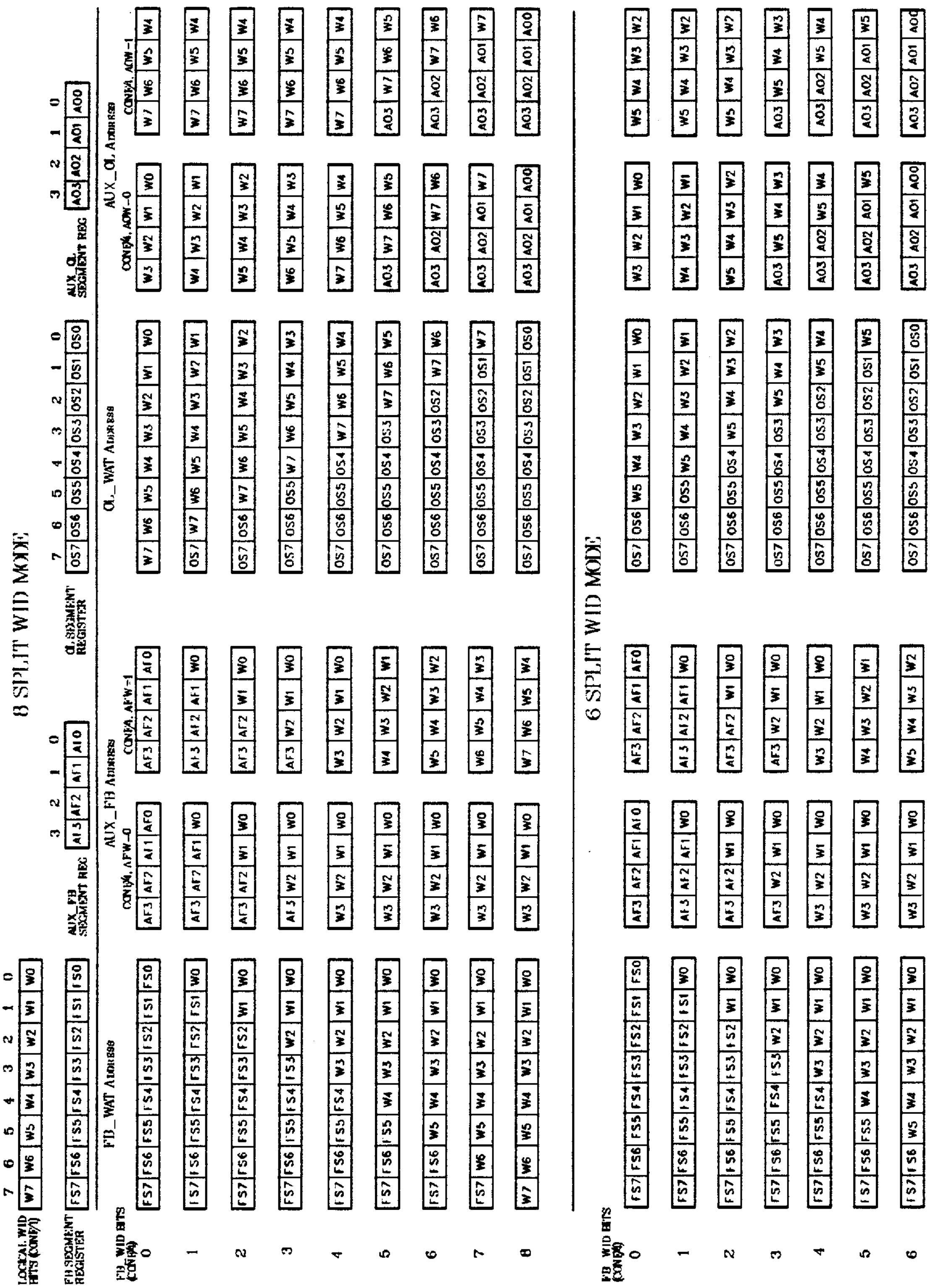


Figure 3. WAT Addressing: Split WID Mode

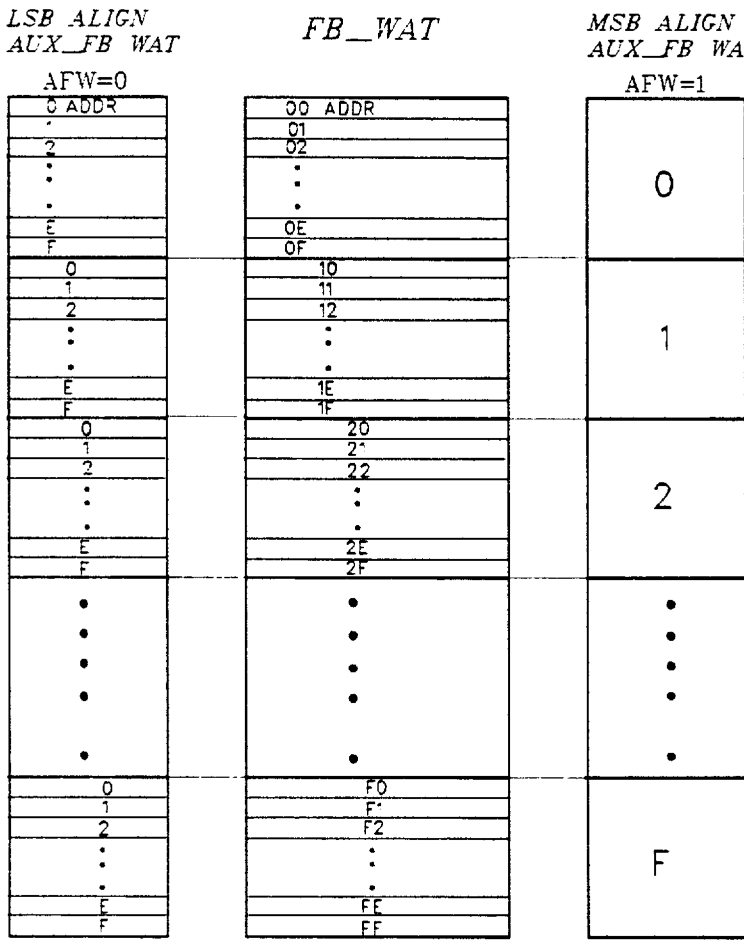


Figure 4. AUX_FB_WAT Addressing Alignment to FB_WAT. Same alignment for AUX_OL_WAT to OL_WAT

Pixel Formats

Multiple pixel color modes are supported by the RGB561. The following table shows graphically how the different pixel color modes are processed.

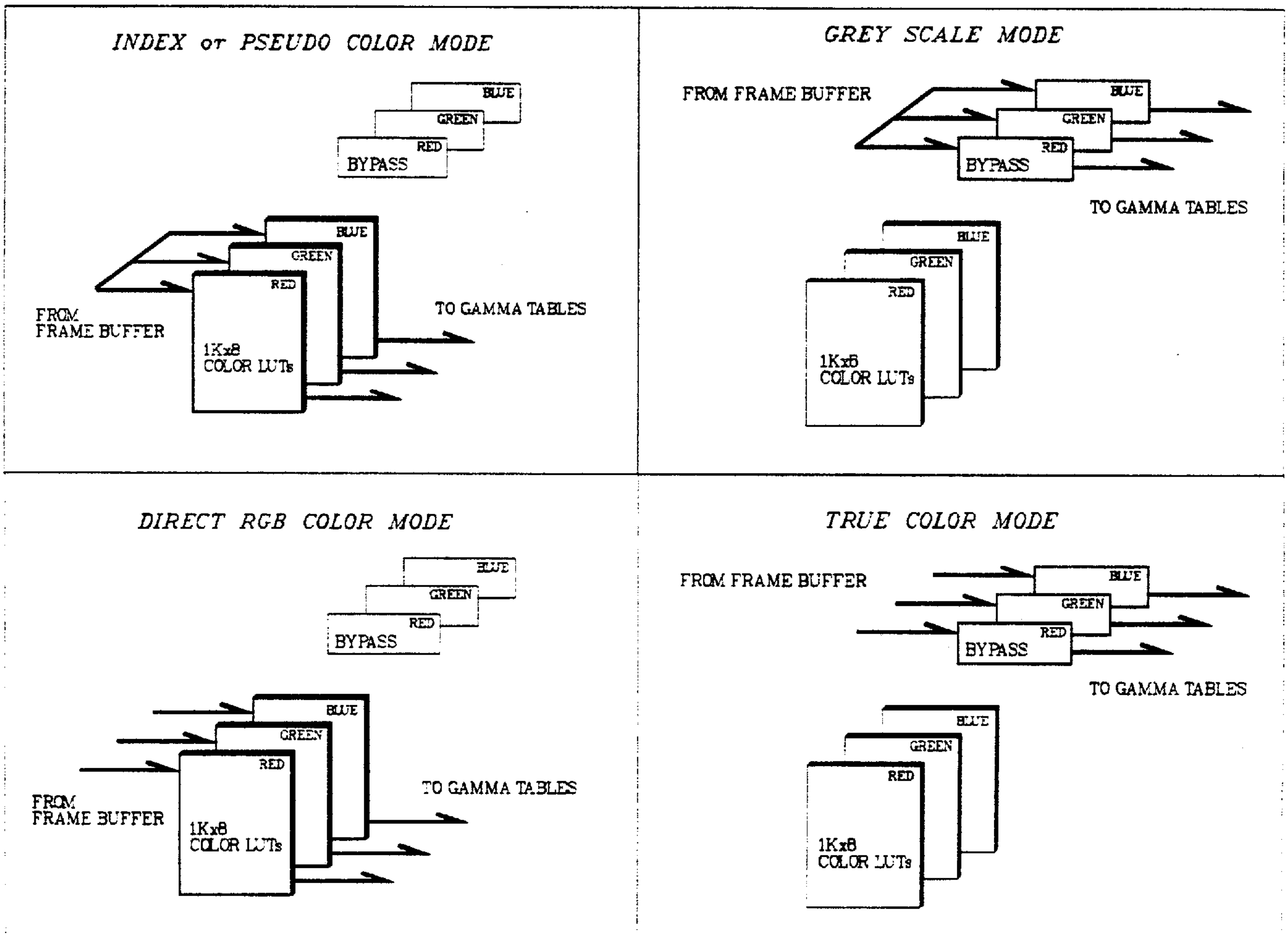


Figure 5. Pixel Color Modes

Pixel format, buffer select, index/direct entries in the FB and OL WATs specify the interpretation of the pixel data on per pixel basis, as specified by the window ID. The starting location of a color LUT within the 1K entry LUT is specified through the WAT (START_ADDR).

Valid pixel formats for the FB, OL and UL layers are shown in the following tables. The color data components for each pixel format are configurable to either RGB or BGR, but are shown here as RGB. Use of pixel formats not supported by a selected serializer configuration or use of invalid pixel formats will produce unpredictable results.

The overlay port is further divided into overlay and underlay layers. In pixel formats where underlay is not provided, all WAT entries pertaining to the underlay layer are ignored.

Table 16. RGB561 Pixel Data Formats

For the color modes below, frame buffer data locations are specified in RGB format for frame buffers A/B.

24 bpp		RED_B 40	$GREEN_B$ 32	$BLUE_B$ 24	RED_A 16	$GREEN_A$ 8	$BLUE_A$ 0
16 bpp		R_B 27	G_B 21	B_B 16	R_A 11	G_A 5	B_A 0
12 bpp		R_B 20	G_B 16	B_B 12	R_A 8	G_A 4	B_A 0
8 bpp		R_B13	G_B10	B_B8	R_A5	G_A2	B_A0
8 bpp						$INDEX_B$ 8	$INDEX_A$ 0

For the overlay (OL) and underlay (UL) partitioning specified, data locations are shown for an RGB format for frame buffers A/B. Anything less than 8 bpp is considered an index mode.

Table 17. Overlay/Underlay Formats

8b. DB	R_B13	G_B10	B_B8	R_A5	G_A2	B_A0
8b		$INDEX_B$ 8			$INDEX_A$ 0	
6b-2b		OL_B 10	UL_B		OL_A 2	UL_A
4b-4b		OL_B 12	UL_B 8		OL_A 4	UL_A 0
4b					OL_B 4	OL_A 0

The RGB pixel or the index pixel can either be applied to the LUT or bypassed around the LUT, as specified by the WAT color mode (MODE).

The RGB components or the index data as shown in the left column of Table 18 is added to LUT Start Address to produce an offset to the address as shown in Table 1 on page 2.

In the LUT bypass case, the pixel data is linearized to produce an 8 bit result. The linearization result is shown in the right column of Table 18 on page 21.

9006146 0000239 405

Table 18. Color LUT Bypass Conversion Table

LUT ADDRESS INPUT								LUT BYPASS OUTPUT							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
2 bit INDEX Mode to 8 bit Grey scale • Same values are used for RGB															
0	0	0	0	0	0	I1	I0	INVALID							
4 bit INDEX Mode to 8 bit Grey scale with bypass • Same values are used for RGB															
0	0	0	0	I3	I2	I1	I0	I3	I2	I1	I0	I3	I2	I1	I0
6 bit INDEX Mode to 8 bit Grey scale • Same values are used for RGB															
0	0	I5	I4	I3	I2	I1	I0	INVALID							
8 bit INDEX Mode to 8 bit Grey scale • Same values are used for RGB															
I7	I6	I5	I4	I3	I2	I1	I0	I7	I6	I5	I4	I3	I2	I1	I0
8 bit Direct RGB Mode to 24 bit True Color															
0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	R2 G2 0	R1 G1 B1	R0 G0 B0	R2 G2 B1	R1 G1 B0	R0 G0 B1	R2 G2 B0	R1 G1 B1	R0 G0 B0	R2 G2 B1	R1 G1 B0
12 bit Direct RGB Mode to 24 bit True Color															
0 0 0	0 0 0	0 0 0	0 0 0	R3 G3 B3	R2 G2 B2	R1 G1 B1	R0 G0 B0	R3 G3 B3	R2 G2 B2	R1 G1 B1	R0 G0 B0	R3 G3 B3	R2 G2 B2	R1 G1 B1	R0 G0 B0
16 bit Direct RGB Mode to 24 bit True Color															
0 0 0	0 0 0	0 G5 0	R4 G4 B4	R3 G3 B3	R2 G2 B2	R1 G1 B1	R0 G0 B0	R4 G5 B4	R3 G4 B3	R2 G3 B2	R1 G2 B1	R0 G1 B0	R4 G0 B4	R3 G5 B3	R2 G4 B2
24 bit Direct RGB Mode to 24 bit True Color															
R7 G7 B7	R6 G6 B6	R5 G5 B5	R4 G4 B4	R3 G3 B3	R2 G2 B2	R1 G1 B1	R0 G0 B0	R7 G7 B7	R6 G6 B6	R5 G5 B5	R4 G4 B4	R3 G3 B3	R2 G2 B2	R1 G1 B1	R0 G0 B0

Overlay/Underlay Pixel Data

The OL_WAT specifies how the OL pixel bits are to be partitioned between overlay and underlay data. The combinations for the various color modes are shown in Table 17 on page 20. If overlay consists of 8 bit data, it can be interpreted as either an index into the OL_WAT, as direct RGB or as true color. Overlay and underlay combinations with fewer than 8 bits are considered an index. The Color LUTs for overlay and underlay are specified by the Color LUT Start Address in the OL_WAT.

Overlay Transparency

Overlay transparency is determined by the data stored in the AUX_OL_WAT, CK/OT bits, specifying the value to be compared with overlay data. If they match, overlay is transparent. The following choices for comparison are available.

- if overlay data is equal to X'00
- if overlay data is equal to X'FF
- if overlay data is equal to the value stored in ***Chroma Key 0 Register***, masked by ***Chroma Key 0 Mask Register***
- if overlay data is equal to the value stored in ***Chroma Key 1 Register***, masked by ***Chroma Key 1 Mask Register***

Cursor Functions

Cursor/Cross-Hair Features

Cursor

The RGB561 cursor is a 64x64 pixel map (icon) with programmable color and placement in display coordinate space. Each pixel in the cursor is represented by 2 bits of data, these bits are used to determine the cursor color as described earlier in the cursor LUT section. The cursor location refers to the X/Y coordinate of the cursor's programmable ***HOT SPOT***.

Cursor Location

The ***Cursor Hot Spot Registers*** identify the X/Y coordinate, within the 64x64 cursor area, used to position the cursor on the screen (the hot spot). Coordinate (0,0) is the upper left corner of the cursor.

The cursor hot spot is positioned on the screen using the X/Y screen coordinates stored as 16b - two's complement values in the ***Cursor X/Y High/Low Registers***. Valid values range from -256 to 2047 (X'FF00 - 07FF). Screen coordinate (0,0) is the top left screen pixel. The cursor will be moved after the Cursor Y High register is written; it should be the last cursor position register updated or undesirable cursor movement may be visible on the display.

Updates to the cursor position can be made asynchronously or synchronized to occur during vertical blanking, this is controlled by CONF/3, CUC (Cursor Update Control).

Cursor Blinking

The cursor can be made to blink by two different methods. It can blink by switching between two different cursor colors or it can switch between the cursor color and transparency.

The 8 bit ***Blink Rate Register*** contains a value one less than the number of VSYNC pulses in one blink cycle. The 8 bit ***Blink Duty Cycle Register*** contains one less than the number of VSYNC pulses, within the blink cycle, defining the length of time that primary colors from the cursor LUT are displayed. The remaining portion of the blink cycle will select the blink colors from the cursor LUT or force cursor transparency. The Cursor Blink Rate must be greater than the Duty Cycle for blinking to occur. The Cursor Control Register enables independent blinking for the cursor or cross-hair and also specifies the blinking parameters.

Cross-Hair

The cross-hair is a full screen cursor resource that can be used independently of, or in concert with the cursor pixel map. It can be up to 7 pixels wide in the vertical arm and up to 7 lines wide in the horizontal arm. The cross-hair with is programmed to be 1, 3, 5 or 7 pixels in each direction. The center of the intersection of the cross-hair arms is used for positioning.

The cross-hair has two views, default and extended. The default is monochrome with programmable width and color. Extended view has border, outline and fill patterns each with programmable width and color.

Cross-Hair/Cursor Lock

The cross-hair can be 'locked' to the cursor (Cursor Control Register, SC), in this mode they share the cursor location registers and color LUTs. When 'un-locked' the cross-hair has a unique color LUT and location registers.

If the cross-hair is 'unlocked' after being 'locked' to the cursor, the cross-hair position registers are set to the last cursor position and the cross-hair is 'parked' at this position. If the cursor is subsequently moved to a different location and the cross-hair is 'locked' to the cursor again the cross-hair location registers will be updated to the current cursor location and 'snap' the cross-hair to the cursor location.

Cross-Hair Location

The location of the Cross-Hair on the screen is stored in the ***Cross-Hair X/Y High/Low Registers***. The X/Y location specified is the center of the cross-hair arms intersection.

When locked with the cursor, the cursor location registers are used to position both the cross-hair and the cursor, updates to the cross-hair position registers through the MPI are ignored. The Cross-Hair will be moved after the Cross-Hair Y High register is written; it should be the last cross-hair position register updated or undesirable cross-hair movement may be visible on the display.

Updates to the cross-hair position can be made asynchronously or synchronized to occur during vertical blanking. This is controlled by CONF/3, CUC (Cursor Update Control).

Cross-Hair Scissor Clipping

The cross-hair can be clipped to a rectangular area by specifying the upper left screen coordinate in the ***Scissor Start X/Y High/Low Register*** and the lower right screen coordinate in the ***Scissor End X/Y High/Low Register***. The clipping window is applied after the ***Scissor End (Y) High Register*** is written with valid values from 0 to 4095.

Updates to the clipping window coordinates can be made asynchronously or synchronized to occur during vertical blanking. This is controlled by CONF/3, CUC (Cursor Update Control).

Cross-Hair Window Clipping

The cross-hair can be clipped to a logical window by use of the WID clip enable bit in the Cursor control register and appropriate entries in the ***AUX_FB_WAT*** or ***AUX_OL_WAT*** cross-hair enable bits. The resulting clipping window is the logical AND of the scissor clipping area and the window clipping area.

Extended View Cross-Hair

The extended cross-hair pattern contains fill, border and outline colors in an area of up to 7 pixels in both the vertical and horizontal directions. The ***Pattern Color Register*** stores a 2 bit LUT index for each pattern component (00 is transparent).

Each bit in these pattern registers defines the color areas for each of the 7 vertical and 7 horizontal pixels that make up the extended cross-hair pattern. Bit 3 is used as the center of the cross-hair cursor for screen placement. The Cross-hair Control Register, XIP bit determines the color priority at pattern intersections, WIDTH sets the cross-hair width, and EP enables the extended pattern. The areas are determined as follows:

BORDER Bits 6/0 start the top/bottom (vertical pattern) or left/right (horizontal pattern) pixel border areas which are set to 0 to display the border color. The border width is set by placing 0's in register locations while proceeding to the center of the cross-hair (Bit 3).

OUTLINE A 1 in the register indicates an outline color area.

FILL The fill area is displayed for all pixel register location of 0, bounded by outline 1 areas.

When the cross-hair width is less than 7 pixels, determine the color areas specified by the 7b register entry, then, starting from Bit 3, grow the cursor to the specified width.

Cursor / Cross-Hair Interaction

The cursor and cross-hair have independent enable and blinking control. When enabled, the blinking of one cursor can affect the the color of the other at points of intersection.

There is programmable priority/color mix when the cursor and cross-hair intersect; OR, XOR, cursor priority and cross-hair priority are available choices.

Display Priority

Pixel Priority

Pixel information is displayed in the following priority.

- The CURSOR information has the highest priority unless the CURSOR and the Cross-Hair intersect. When they intersect, their relative priority is defined in the Cursor Control register.
- The Cross-Hair in an overlay window has the next highest priority. If the overlay is transparent, the overlay Cross-Hair is considered disabled.
- Information on the overlay port has the next highest priority.
- The Cross-Hair in a frame buffer window has the next highest priority.
- Frame buffer data is the next highest priority.
- The lowest priority to be displayed is underlay data. If there is no valid underlay data or underlays are disabled, then the frame buffer data is forced to be opaque.

Note

The next lower priority level will be displayed if an item is disabled or transparent for the current pixel.

Chroma Key

Chroma keying is a technique to selectively merge two images. A transparency value is stored in the maskable chroma key register that is compared with overlay data. If they match, the overlay data is transparent and the underlying pixel data is displayed. In the RGB561 two chroma key registers are available, each with a corresponding mask register to identify bits in the chroma key register to be used for comparison. Two fixed values (X'00, FF) are also available for determination of overlay transparency.

Chroma keying is enabled by the CK/OT bits in the ***AUX_OL_WAT*** which is addressed by the OL WID bits.

The two 8-b **Overlay Chroma Key Registers** contain an arbitrary transparency value from 0 to 255, to which the overlay data is compared.

Each Chroma Key register has a corresponding ***Overlay Chroma Key Mask Register*** that identifies the bits in the Chroma Key Register which are to be used for comparison with overlay data in determining overlay transparency. A Mask Register bit set to 1 selects the corresponding bit in the ***Overlay Chroma Key Register***.

Synchronization

The RGB561 synchronizes internal functions to the incoming composite blank and composite sync signals. The user has the choice of an externally supplied pixel clock or internal PLL generated pixel clock. The RGB561 is also capable of generating the Serial-Clock if desired.

Note: For proper synchronization the Vertical portion of the composite blank signal (CBLANKIN) must be active for at least 256 LOAD_CLK cycles.

Control Register

The SYNC control register is used to control the HSYNC chip output. It also controls SYNC-on-GREEN and the blanking pedestal for the DAC.

X'0020 SYNC Control Register

The synchronizing signals and their respective functions are summarized in Table 19 and Table 25 on page 50. Their associated timings are found in "▪AC Characteristics" on page 55.

Table 19. DTG Signal Summary

Signal	IO	Function
HSYNC	O	The CSYNC input, delayed by the chip pipeline, is output on this pin
CSYNC	I	CSYNC or HSYNC input from the controller
CBLANKIN	I	Composite blank from Controller
CBLANKOUT	O	Timing reference for VIDEO and DAC outputs
FIELD	I	Controller input for even/odd scan lines in interlaced modes
LOAD_CLK	I	Free running Load Clock from controller
SERIAL_CLK	O	Controlled by SC bit in CONF/3
AUX_SERIAL_CLK	O	Controlled by SC bit in CONF/3

Clocking

Pixel Clock

Clocking can be taken from an external pixel clock applied at **EXTCLK/$\overline{\textbf{EXTCLK}}$** inputs or from the on-chip PLL.

Phase Locked Loop

The RGB561 incorporates a flexible PLL capable of providing pixel clock rates up to 250 MHz from a low frequency reference. The PLL requires a reference from a stable frequency source of 4-100 MHz, applied on the ***REFCLK*** input. Alternatively the PLL can be locked to the clock signal applied on the **EXTCLK/$\overline{\textbf{EXTCLK}}$** ECL inputs as selected in the CONF/2 register.

The PLL is programmed with two registers; the ***PLL Reference Register*** and the ***PLL/VCO Divider Register***. The reference register is used to pre-scale the reference frequency for the PLL phase detector. The VCO divider register is used to set the desired pixel clock frequency range (DFR) and the actual pixel clock frequency (VF).

For best results, it is imperative that the guidelines for wiring and placing the external PLL components be followed as specified in "Circuit Schematic" on page 54. It is also important that the PLL control and operating registers be properly initialized prior to enabling the PLL.

External Pixel Clock

An external pixel clock can be used on the RGB561, the EXTCLK ECL inputs can be selected using the CONF/2 register.

If **EXTCLK/$\overline{\textbf{EXTCLK}}$** inputs are used directly as the pixel clock and the PLL is disabled, the desired frequency range value, ***DFR***, must still be programmed to assure proper look-up table access.

Serial_Clock

The SERIAL_CLK is used to access the VRAM frame buffer data and is synchronized with the LOAD_CLK input, which latches data into the RGB561 serializer. If SERIAL_CLK is provided by the controller, it is expected that LOAD_CLK is derived from it when provided to the RGB561. When SERIAL_CLK is driven by the RGB561, controlled by the SC bit in Configuration Reg. 3, the far end of the signal is used as the LOAD_CLK input.

Auxillary Serial Clock

The AUX_SERIAL_CLK is available for use as a second SERIAL_CLK source.

Divided DOT Clock

The divided dot clock output, ***DDOTCLK***, is a programmable division of the internal pixel clock or, optionally, the SERIAL_CLK may be placed on this output. It can be used as a timing reference for the ***VIDEO*** or RGB outputs. Timing relationships are provided in "▪AC Characteristics" on page 55.

Register Address Table

Table 20. Register Addresses

$ADDR_{15-0}$	C1/C0	REGISTER NAME	pg
	0 0	ADDRESS INDEX REGISTER LOW (7-0)	
	0 1	ADDRESS INDEX REGISTER HIGH (15-8)	
X'0000	1 0	REVISION LEVEL REGISTER (READ ONLY)	28
X'0001	1 0	CONFIGURATION REGISTER 1 (CONF/1)	29
X'0002	1 0	CONFIGURATION REGISTER 2 (CONF/2)	30
X'0003	1 0	CONFIGURATION REGISTER 3 (CONF/3)	31
X'0004	1 0	CONFIGURATION REGISTER 4 (CONF/4)	32
X'0005	1 0	INTERLEAVE CONTROL REGISTER	33
X'0006	1 0	FB_WAT SEGMENT REGISTER	34
X'0007	1 0	OL_WAT SEGMENT REGISTER	34
X'0008	1 0	AUX_FB_WAT SEGMENT REGISTER	34
X'0009	1 0	AUX_OL_WAT SEGMENT REGISTER	34
X'000A - X'000F	—	RESERVED	
X'0010	1 0	OL CHROMA KEY 0 REGISTER	34
X'0011	1 0	OL CHROMA KEY 1 REGISTER	34
X'0012	1 0	OL CHROMA KEY 0 MASK REGISTER	34
X'0013	1 0	OL CHROMA KEY 1 MASK REGISTER	34
X'0014 - X'001F	—	RESERVED	
X'0020	1 0	SYNC CONTROL REGISTER	35
X'0021	1 0	PLL VCO DIVIDER REGISTER	36
X'0022	1 0	PLL REFERENCE REGISTER	36
X'0023 - X'002F	—	RESERVED	
X'0030	1 0	CURSOR CONTROL REGISTER	37
X'0031	1 0	CROSS-HAIR CONTROL REGISTER	39
X'0032	1 0	CURSOR BLINK RATE REGISTER	38
X'0033	1 0	CURSOR BLINK DUTY CYCLE REGISTER	38
X'0034	1 0	CURSOR HOT SPOT X REGISTER	38
X'0035	1 0	CURSOR HOT SPOT Y REGISTER	38
X'0036	1 0	CURSOR (X) LOW REGISTER (7-0)	38
X'0037	1 0	CURSOR (X) HIGH REGISTER (15-8)	38
X'0038	1 0	CURSOR (Y) LOW REGISTER (7-0)	38
X'0039	1 0	CURSOR (Y) HIGH REGISTER (15-8)	38
X'003A - X'003F	—	RESERVED	
X'0040	1 0	SCISSOR START (X) LOW REGISTER (7-0)	40
X'0041	1 0	SCISSOR START (X) HIGH REGISTER (15-8)	40
X'0042	1 0	SCISSOR START (Y) LOW REGISTER (7-0)	40
X'0043	1 0	SCISSOR START (Y) HIGH REGISTER (15-8)	40
X'0044	1 0	SCISSOR END (X) LOW REGISTER (7-0)	40
X'0045	1 0	SCISSOR END (X) HIGH REGISTER (15-8)	40
X'0046	1 0	SCISSOR END (Y) LOW REGISTER (7-0)	40
X'0047	1 0	SCISSOR END (Y) HIGH REGISTER (15-8)	40
X'0048	1 0	CROSS-HAIR (X) LOW REGISTER (7-0)	40
X'0049	1 0	CROSS-HAIR (X) HIGH REGISTER (15-8)	40
X'004A	1 0	CROSS-HAIR (Y) LOW REGISTER (7-0)	40
X'004B	1 0	CROSS-HAIR (Y) HIGH REGISTER (15-8)	40
X'004C	1 0	CROSS-HAIR PATTERN COLOR REGISTER	41
X'004D	1 0	HORIZONTAL CROSS-HAIR PATTERN REGISTER	41
X'004E	1 0	VERTICAL CROSS-HAIR PATTERN REGISTER	41
X'004F	—	RESERVED	
X'0050	1 0	VRAM BIT MASK REGISTER 1	43
X'0051	1 0	VRAM BIT MASK REGISTER 2	43
X'0052	1 0	VRAM BIT MASK REGISTER 3	43
X'0053	1 0	VRAM BIT MASK REGISTER 4	43
X'0054	1 0	VRAM BIT MASK REGISTER 5	43
X'0055	1 0	VRAM BIT MASK REGISTER 6	43
X'0056	1 0	VRAM BIT MASK REGISTER 7	43
X'0057 - X'005E	—	RESERVED	
X'005F	1 0	DAC CONTROL REGISTER	41
X'0060 - X'0063	1 0	MISR REGISTERS (0 - 3) (READ ONLY)	42
X'0064	1 0	DAC COMPARATOR OUTPUT REGISTER (READ)	43
X'0065	1 0	MISR STATUS (READ)	42
X'0066 - X'006F	—	RESERVED	
X'0070 - X'0081	—	RESERVED	
X'0082	1 0	DIVIDED DOT CLOCK REGISTER	44
X'0083 - X'0A0F	—	RESERVED	
X'0A10 - X'0A17	1 1	CURSOR LUT (3 RW/ADDR)	45
X'0A18 - X'0A1F	1 1	CROSS-HAIR CURSOR LUT (3 RW/ADDR)	45
X'0A20 - X'0DFF	—	RESERVED	
X'0E00 - X'0E0F	1 0	AUXILIARY FRAME BUFFER WAT	46
X'0E10 - X'0EFF	—	RESERVED	
X'0F00 - X'0F0F	1 0	AUXILIARY OVERLAY WAT	46
X'0F10 - X'0FFF	—	RESERVED	
X'1000 - X'10FF	1 1	FRAME BUFFER WAT (2 RW/ADDR)	47
X'1100 - X'13FF	—	RESERVED	
X'1400 - X'14FF	1 1	OVERLAY WAT (2 RW/ADDR)	48
X'1500 - X'1FFF	—	RESERVED	
X'2000 - X'23FF	1 0	CURSOR PIXMAP - 1Kx8 SRAM	49
X'2400 - X'2FFF	—	RESERVED	
X'3000 - X'30FF	1 1	RED GAMMA LUT - 256X10 SRAM (2 RW/ADDR)	
X'3100 - X'33FF	—	RESERVED	
X'3400 - X'34FF	1 1	GREEN GAMMA LUT - 256X10 SRAM (2 RW/ADDR)	
X'3500 - X'37FF	—	RESERVED	
X'3800 - X'38FF	1 1	BLUE GAMMA LUT - 256X10 SRAM (2 RW/ADDR)	
X'3900 - X'3FFF	—	RESERVED	
X'4000 - X'43FF	1 1	COLOR LUT - 3x(1Kx8) SRAM (3 RW/ADDR)	
X'4400 - X'7FFF	—	RESERVED	

X'0000 Revision Level Register (Read only)

The Revision level register may be accessed by the controller to determine the vintage and type of RAMDAC in use. RGB561 presently returns X'10 in the latest version.

Configuration Registers

Chip configuration parameters are typically initialized from the MPI port at power on. They specify various operating mode parameters the graphics adapter will be using on a screen basis. Before enabling a function in the configuration registers, the control registers associated with that function should be programmed. The Configuration Registers are CONF/1, CONF/2, CONF/3, CONF/4, and INTERLEAVE.

The complete list of frame buffer input data configurations is shown in Table 7 on page 9. Selections should be consistent with these options or unpredictable results will occur.

The configurations are set in the following registers:

CONF/1 Set serializer multiplexing and number of OL and window ID bits.

CONF/2 Enable VRAM masking, PLL, DTG and RGB outputs to the display; select the reference clock for the PLL or use EXTCLK as the pixel clock.

CONF/3 Enable interlace, MISR; activate SERIAL_CLK and VIDEO outputs; set the FIELD signal polarity, and RGB/BGR input format

CONF/4 Specify the number of FB_WIDs, whether split or common addressing for FB_WAT and OL_WAT, and address alignment for AUX_FB_WAT/AUX_OL_WAT access.

INTERLEAVE Enable interleave mode, set the starting pixel, specify if OL/WID data is to be interleaved with pixel data.

X'0001 Configuration Register 1 (CONF/1)

7	6	5	4	3	2	1	0
MUX			OVLY		LWID		

MUX	***Serializer MUX Mode Select***	
	000	5:1 BASIC
	001	4:1 BASIC
	010	4:1 EXTENDED
	011	4:1 SUPER_EXTENDED
	100	5:1 30 bpp
	101	8:1 MODE B
	110	4:1 30 bpp
	111	8:1 MODE A
OVLY	***Overlay Bits***	
	00	0 bits
	01	8 bits
	10	16 bits
	11	Variable WAT control
LWID	***Logical Window ID Bits***	
	000	0 bits
	001	2 bits
	010	4 bits
	011	6 bits
	1XX	8 bits

Note

When LWID specifies fewer logical Window ID bits than the maximum Window ID bits available, the 8 bit WAT address is formed by using the LWID specified least significant bits from the serialized WID value and padding the missing most significant bits with zeros.

X'0002 Configuration Register 2 (CONF/2)

7	6	5	4	3	2	1	0
0	0	VM	PLL	CLC	FREQ	PIX	SCR

VM	***VRAM Mask Register Control*** *0* Disable VRAM Masking *1* Enable VRAM Masking	VRAM masking is used in conjunction with the MISR for diagnostics and fault isolation. Contents of the mask register are used to set VRAM inputs to 0.
PLL	***Phase Lock Loop Control*** *0* Disable PLL *1* Enable PLL	The PLL should not be enabled until the ***PLL/VCO Divider and PLL Reference Registers*** (X'0021, 0022) have been initialized. When using the external pixel clock, EXTCLK (PIX=1), the PLL should be disabled (PLL=0).
CLC	***Cursor Location Control Enable*** *0* Disable CLC *1* Enable CLC	When CLC=0, updates to the ***Cursor Location Registers, Scissor Location Registers and Cross-Hair Location Registers*** (X'0036-0039, 0040-004B) occur immediately. If CLC=1, updates are made only after the Y High End values are updated. This bit should be set to zero for register diagnostics of pending update values.
FREQ	***PLL Reference Frequency Select*** *0* REFCLK is reference *1* EXTCLK is reference	This bit selects the PLL reference frequency as either the REFCLK input or the EXTCLK/$\overline{\text{EXTCLK}}$ inputs, at frequencies between 4 and 62 MHz. If the PIX bit has selected EXTCLK/$\overline{\text{EXTCLK}}$ as the external timing source, this bit is ignored.
PIX	***Pixel Clock Timing Select*** *0* Use on-chip PLL *1* Use EXTCLK/$\overline{\text{EXTCLK}}$ (PLL=0)	This bit activates the external pixel clock inputs, EXTCLK/$\overline{\text{EXTCLK}}$, to be used as the internal pixel clock.
SCR	***Screen Control*** *0* Disable RGB Outputs *1* Enable RGB Outputs	This forces the DAC outputs to the blanking level for use during diagnostic mode. SYNCs are still on the GREEN DAC if composite SOG is enabled. The ***DAC Control Register*** (X'005F) DAC bit can be used to power the DAC off for VIDEO mode.

X'0003 Configuration Register 3 (CONF/3)

7	6	5	4	3	2	1	0
IC	SC	FP	MISR	CUC	0	VID	RGB

IC	***Interlace Control*** *0* Non-Interlaced Mode *1* Interlaced Mode	Interlaced modes require the FIELD signal which indicates an even or odd scan line for proper cursor and interleave outputs.
SC	***SERIAL_CLK Control*** *0* Tristate outputs *1* Enable outputs	The SERIAL_CLK is used as VRAM serial clock and as LOAD_CLK for frame buffer inputs. The AUX_SERIAL_CLK can be used as a second SERIAL_CLK source for loading reasons or used with RGB061 when VRAMs are dotted on the frame buffer interface to control buffer selection. When used as a second SERIAL_CLK the CYCLES value in the AUX_SERIAL_CLK_CONTROL register must be zero.
FP	***Field Polarity Select*** *0* 0=EVEN, 1 ODD Scan Line *1* 1=EVEN, 0 ODD Scan Line	This bit selects the FIELD input polarity for ODD/EVEN lines. It is used to produce the correct cursor and interleave data. The appropriate VRAM data must be presented.
MISR	***Diagnostic MISR Run Control*** *0* Disable MISR *1* Enable MISR	Enabling resets the MISR register to X'3FFFFFFF for frame signature accumulation when vertical blanking becomes inactive. The screen can be blanked while running test frames with CONF/2, SCR.
CUC	***Cursor/Scissor Update Control*** *0* Synchronous Updates *1* Asynchronous Updates	Synchronous cursor/scissor position updates are made at the end of VSYNC during the VBLANK period that follows the Y High End Position Register update. Asynchronous clipping updates occur immediately after writing the Y High End register.
VID	***VIDEO Output Control*** *0* Disable VIDEO output *1* Enable VIDEO output	The signal timings associated with this mode are shown in Figure 8 on page 56. When the VIDEO port is enabled output is provided as described in "Video" on page 5. The MPI is available during blanking periods for register access.
BIT 2	***Reserved***	Reserved bit. Must be set to 0 for proper operation.
RGB	***RGB/BGR Color Format*** *0* BGR Pixel format *1* RGB Pixel format	The figures in this document depict RGB mode. RED and BLUE data locations are exchanged in BGR format. This bit does not change the MPI update data sequence used to load the color LUTs, which is always RGB.

X'0004 Configuration Register 4 (CONF/4)

7	6	5	4	3	2	1	0
0	FB_WID				SWE	AOW	AFW

FB_WID	***# of FB WID bits*** ***0000*** 0 FB_WID Bits ***0001*** 1 FB_WID Bits ***0010*** 2 FB_WID Bits ***0011*** 3 FB_WID Bits ***0100*** 4 FB_WID Bits ***0101*** 5 FB_WID Bits ***0110*** 6 FB_WID Bits ***0111*** 7 FB_WID Bits ***1xxx*** 8 FB_WID Bits	This register specifies parameters for Window Attribute Addressing which is described in "WAT Addressing" on page 15. The FB_WID bits specify the number of WID bits (CONF/1, WID) to be allocated as FB_WID bits for WAT addressing. The FB_WID bits are taken from the WID LSBs, any remaining WID bits may be used as OL_WID bits taken from the most significant WID bits. The selected bits are placed in the LSB bit positions of the FB_WAT or OL_WAT address with any additional address bits being taken from the appropriate ***FB_WAT or OL_WAT Segment Registers*** (X'0006-0009). FB_WID is only valid in split WID mode (SWE=1). If X'0000 is selected, the FB_WAT address is taken from the FB_WAT Segment Register to process all frame buffer data. If the number FB_WID bits = the total WID bits available, none remain for the OL_WID Address and the OL_WAT Segment Register data is used to access the OL_WAT and process all overlay/underlay data.
SWE	***Split WID Enable*** ***0*** WID bits common for FB/OL ***1*** WID bits split between FB/OL	SWE specifies the WID bits to be common and used to address both the FB_WAT and the OL_WAT, or split between them. In split mode, OL_WIDs are the difference between the total WID bits (CONF/1) and the FB_WID bits.
AOW	***AUX_OL_WAT Address Alignment*** ***0*** Use 4 LSBs of OL_WID ***1*** Use 4 MSB of OL_WID	Specifies either the least or most significant 4 OL_WID bits to be used to form the AUX_OL_WAT Address. Refer to "WAT Addressing" on page 15 for details.
AFW	***AUX_FB_WAT Address Alignment*** ***0*** Use 4 LSBs of FB_WID ***1*** Use 4 MSbs of FB_WID	Specifies either the least or most significant 4 FB_WID bits to be used to form the AUX_FB_WAT Address. Refer to "WAT Addressing" on page 15 for details.

X'0005 Interleave Control Register

Refer to "Pixel Interleave" on page 13 for definition of interleave modes. Table 7 on page 9 specifies the configuration modes which allow interleave of OL/UL and WID data with pixel data.

7	6	5	4	3	2	1	0
ILVE			PIXEL			OUS	WIE

ILVE	***INTERLEAVE Mode Select***
	000 Mode 0
	001 Mode 1
	010 Mode 2
	011 Mode 3
	100 Mode 4
	101 Mode 5
	110 Mode 6
	111 Mode 7
PIXEL	***1st Scan Line Pixel***
	000 PIXEL A
	001 PIXEL B
	010 PIXEL C
	011 PIXEL D
	100 PIXEL E
	101 PIXEL F
	110 PIXEL G
	111 PIXEL H
OUS	***OL / UL Interleave Enable***
	0 Disabled
	1 Interleave OL/UL and pixel data
WIE	***WID Interleave Enable***
	0 Disabled
	1 Interleave WID and pixel data

X'0006-0007 WAT Segment Registers

7	6	5	4	3	2	1	0
FS_7	FS_6	FS_5	FS_4	FS_3	FS_2	FS_1	FS_0
OS_7	OS_6	OS_5	OS_4	OS_3	OS_2	OS_1	OS_0
X'0006 FB WAT SEGMENT REGISTER							
X'0007 OL WAT SEGMENT REGISTER							

X'0008-0009 AUX WAT Segment Registers

7	6	5	4	3	2	1	0
0	0	0	0	AF_3	AF_2	AF_1	AF_0
0	0	0	0	AO_3	AO_2	AO_1	AO_0
X'0008 AUX FB WAT SEGMENT REGISTER							
X'0009 AUX OL WAT SEGMENT REGISTER							

X'0010-0011 Chroma Key Registers

7	6	5	4	3	2	1	0
CHROMA KEY 0							
CHROMA KEY 1							
X'0010 OL CHROMA KEY 0 REGISTER							
X'0011 OL CHROMA KEY 1 REGISTER							

X'0012-0013 Chroma Key Mask Registers

7	6	5	4	3	2	1	0
MASK VALUE 0							
MASK VALUE 1							
X'0012 OL CHROMA KEY 0 MASK REGISTER							
X'0013 OL CHROMA KEY 1 MASK REGISTER							

9006146 0000253 885

X'0020 SYNC Control Register

7	6	5	4	3	2	1	0
0	0	HOE	0	HPC	0	SOG	BPE

BIT 6	***Reserved***	Reserved bit. Must be set to 0 for proper operation.
HOE	***HSYNC OCD Enable Override*** *0* HSYNC output disabled *1* HSYNC output enabled	Overrides SOG and places SYNC on HSYNC output pin.
BIT 4	***Reserved***	Reserved bit. Must be set to 0 for proper operation.
HPC	***HSYNC Polarity Control*** *0* HSYNC output active low *1* HSYNC output active high	0 selects SYNC normally high with a negative sync pulse, a 1 selects SYNC normally low with a positive sync pulse. This bit controls SYNC polarity when composite SYNC is placed on the HSYNC output pin.
BIT 2	***Reserved***	Reserved bit. Must be set to 0 for proper operation.
SOG	***Composite SYNC-on-GREEN*** *0* Composite SOG disabled *1* Composite SOG enabled	Composite sync can be placed on the GREEN DAC output (SOG=1) which will tri-state the HSYNC output, unless the HSYNC output is enabled by HOE bit. SOG is independent of the polarity specified by HPC.
BPE	***Blanking Pedestal Enable*** *0* Pedestal disabled *1* Pedestal enabled	The blanking pedestal provides different voltage levels for black and blank on the DAC outputs as specified in Figure 9 on page 59 and Figure 10 on page 59.

PLL Programming

The RGB561 PLL is implemented with an $\frac{M}{N \times L}$ architecture.

The general PLL programming equations follow:

$$f_{PLL} = \frac{M \times f_{ref}}{2 \times N \times L} \text{ for } 16.25 \le f_{PLL} \le 128 \text{ Mhz.}$$

with $65 \le M \le 128$ and $2 \le N \le 31$ and $L = 1,2,4$.

$$f_{PLL} = \frac{M \times f_{ref}}{N \times L} \text{ for } 128 < f_{PLL} \le 256 \text{ Mhz.}$$

with $65 \le M \le 128$ and $2 \le N \le 31$ and $L = 1$.

f_{ref} is the ***REFCLK*** input frequency or optionally the ***EXTCLK/$\overline{\text{EXTCLK}}$*** input frequency if used as the PLL reference frequency. The reference frequency must be in the following range, $4 \le f_{ref} \le 100$ Mhz.

X'0021 PLL/VCO Divider Register

The PLL/VCO Divider register contains the M and L values for PLL programming.

7	6	5	4	3	2	1	0
PFR		$M - 65$					

PFR	***PLL Frequency Range*** ***00*** $L = 4$, $16.25 \le f_{PLL} \le 32Mhz$ ***01*** $L = 2$, $32.50 \le f_{PLL} \le 64Mhz$ ***10*** $L = 1$, $65.00 \le f_{PLL} \le 128Mhz$ ***11*** $L = 1$, $128 < f_{PLL} \le 256Mhz$	This selects the range within which the video frequency (VF) falls and at which the PLL will operate. PFR must be programmed even if the external pixel clock inputs (EXTCLK/$\overline{\text{EXTCLK}}$) are being used and the PLL is disabled or LUT accesses will not work properly.

X'0022 PLL Reference Register

The PLL Reference Register contains the N value for PLL programming.

7	6	5	4	3	2	1	0
0	0	0	N				

Cursor Registers

X'0030 Cursor Control Register

7	6	5	4	3	2	1	0
BT	SC	LOP		XB	XE	CB	CE

BT	***Blink to transparent*** ***0*** Blink color from Cursor LUT ***1*** Blink color transparent	Specifies blinking of the cursor between the primary and blink Cursor color, or between the primary cursor color and transparency.
SC	***Separate Cursor Enable*** ***0*** Cursor/X-Hair locked ***1*** Cursor/X-Hair separate	SC enables the CROSS-HAIR to be moved independently of the CURSOR. The default power on mode is to have the CROSS-HAIR and CURSOR registers locked together and updates made to the CURSOR location registers are loaded into the ***Cross-Hair X/Y Location Registers***. SC=1 enables loading of the CROSS-HAIR registers independently of the CURSOR location registers. This bit should be set to a logical 1 for register diagnostics.
LOP	***Overlap Logical Operator*** ***00*** XOR Cursor and X-Hair ***01*** OR Cursor and X-Hair ***10*** Cursor has priority ***11*** X-Hair has priority	When the cross-hair and cursor overlap, 4 display choices are provided at the points of intersection. The output of the logical operators and cursor priority access a color from the Cursor LUT. Cross-hair (X-Hair) priority accesses colors in the Cross-Hair LUT.
XB	***Cross-Hair Blinking*** ***0*** Disabled ***1*** Enabled	Cross-Hair blinking is enabled by XB, the alternating colors are specified by BT and the blink rate is programmed in the ***Blink Rate and Duty Cycle Registers***.
XE	***Cross-Hair Enabling*** ***0*** Disabled ***1*** Enabled	When enabled the cross-hair will be displayed using colors in the Cursor or Cross-Hair LUT depending on the cursor configuration selected. Refer to "Cursor/Cross-Hair Look-Up Tables" on page 3 for details.
CB	***Cursor Blinking*** ***0*** Disabled ***1*** Enabled	Cursor blinking is enabled by CB, alternating colors are specified by BT and the blink rate is programmed in the ***Blink Rate and Duty Cycle Registers***.
CE	***Cursor Enabling*** ***0*** Disabled ***1*** Enabled	The cursor contents stored in the on chip 64x64x2 map is used to select a color from the Cursor LUT, described in "Cursor/Cross-Hair Look-Up Tables" on page 3.

X'0032-0033 Cursor Blink Registers

7	6	5	4	3	2	1	0
VSYNC Pulses (-1) than length of Blink Cycle							
VSYNC Pulses (-1) than primary length to Cursor/Cross-Hair LUT							
X'0032 CURSOR BLINK RATE REGISTER ***X'0033 CURSOR BLINK DUTY CYCLE REGISTER***							

X'0034-0035 Cursor Hot Spot Location Registers

7	6	5	4	3	2	1	0
0	0	X_5	X_4	X_3	X_2	X_1	X_0
0	0	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
X'0034 CURSOR HOT SPOT X REGISTER ***X'0035 CURSOR HOT SPOT Y REGISTER***							

X'0036-0039 Cursor Location Registers

7_H	6_H	5_H	4_H	3_H	2_H	1_H	0_H	7_L	6_L	5_L	4_L	3_L	2_L	1_L	0_L
X_{15}	X_{14}	X_{13}	X_{12}	X_{11}	X_{10}	X_9	X_8	X_7	X_6	X_5	X_4	X_3	X_2	X_1	X_0
Y_{15}	Y_{14}	Y_{13}	Y_{12}	Y_{11}	Y_{10}	Y_9	Y_8	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
X'0037 CURSOR X HIGH REG (15-8) ***X'0039 CURSOR Y HIGH REG (15-8)***								***X'0036 CURSOR X LOW REG (7-0)*** ***X'0038 CURSOR Y LOW REG (7-0)***							

Bits $7\text{-}0_H/7\text{-}0_L$	16b two's complement number with a valid range of -256 to +2047
Bit 7_H	Sign bit
Bits $6\text{-}3_H$	Sign extended internally, writes to these bits are ignored. Read of these bits return the sign value.

X'0031 Cross-Hair Control Register

7	6	5	4	3	2	1	0
XIP	WIDTH		CLIP		COLOR		EP
X'0031 CROSS-HAIR CONTROL REGISTER							

XIP	***X-Hair Intersect Priority (EP=1)*** ***0*** OUTLINE Color Priority ***1*** FILL Color Priority	Intersect priority specifies the color displayed when the outline and fill areas intersect in the Extended Pattern cross-hair (EP=1). The BORDER color has the lowest priority.
WIDTH	***X-Hair Default Width (EP=0/1)*** ***00*** 1 Pixel ***01*** 3 Pixels ***10*** 5 Pixels ***11*** 7 Pixels	This entry sets the width of the non-patterned monochrome cross-hair (EP=0), or the Extended Pattern Cross-hair (EP=1). In the Extended Pattern mode the ***Vertical/Horizontal Pattern Registers*** define the 3 area pattern placement: border, fill and outline.
CLIP	***Cross-Hair Clipping Modes*** ***00*** No Clipping ***01*** Scissor Registers ***10*** Window Coordinates ***11*** Scissor/Window Intersection	The cross-hair can be clipped to a window, using WID bits, to coordinates specified in the ***Cross-hair Scissor Start/End Registers*** or at the intersection of the window and scissor coordinates.
COLOR	***Cross-Hair Color (EP=0)*** ***00*** Transparent (no X-Hair) ***01*** Color 1 ***10*** Color 2 ***11*** Color 3	The COLOR bits are used as an index into the Cursor LUT to select a cross-hair cursor color for the monochrome, non-patterned cross-hair.
EP	***Extended X-Hair Pattern Enable*** ***0*** Disable Pattern registers ***1*** Enable Pattern registers	The extended cross-hair pattern areas have WIDTH, BORDER, OUTLINE and FILL colors (X'004C) and vertical/horizontal pattern selection (X'004D, 004E). When EP=0 a monochrome X-Hair is produced.

X'0040-0047 Cross-Hair Scissor Clipping Registers

7_H	6_H	5_H	4_H	3_H	2_H	1_H	0_H	7_L	6_L	5_L	4_L	3_L	2_L	1_L	0_L
—	—	—	—	X_{11}	X_{10}	X_9	X_8	X_7	X_6	X_5	X_4	X_3	X_2	X_1	X_0
—	—	—	—	Y_{11}	Y_{10}	Y_9	Y_8	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
X'0041 SCISSOR START X HIGH REG (15-8) *X'0043 SCISSOR START Y HIGH REG (15-8)*								*X'0040 SCISSOR START X LOW REG (7-0)* *X'0042 SCISSOR START Y LOW REG (7-0)*							
X'0045 SCISSOR END X HIGH REG (15-8) *X'0047 SCISSOR END Y HIGH REG (15-8)*								*X'0044 SCISSOR END X LOW REG (7-0)* *X'0046 SCISSOR END Y LOW REG (7-0)*							
Bits $3\text{-}0_H/7\text{-}0_L$ Upper left X/Y screen coordinate															

X'0048-004B Cross-Hair Location Registers

7_H	6_H	5_H	4_H	3_H	2_H	1_H	0_H	7_L	6_L	5_L	4_L	3_L	2_L	1_L	0_L
X_{15}	X_{14}	X_{13}	X_{12}	X_{11}	X_{10}	X_9	X_8	X_7	X_6	X_5	X_4	X_3	X_2	X_1	X_0
Y_{15}	Y_{14}	Y_{13}	Y_{12}	Y_{11}	Y_{10}	Y_9	Y_8	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
X'0049 CROSS-HAIR X HIGH REG (15-8) *X'004B CROSS-HAIR Y HIGH REG (15-8)*								*X'0048 CROSS-HAIR X LOW REG (7-0)* *X'004A CROSS-HAIR Y LOW REG (7-0)*							
Bits $7\text{-}0_H/7\text{-}0_L$ 16b two's complement number with a valid range of -256 to +2047 **Bit 7_H** Sign bit **Bits $6\text{-}3_H$** Sign extended internally, writes to these bits are ignored. Read of these bits return the sign value.															

X'004C-004E Cross-Hair Pattern Registers

The cross-hair pattern is described in "Extended View Cross-Hair" on page 23, the ***Cross-Hair Control Register*** is discussed in "X'0031 Cross-Hair Control Register" on page 39.

7	6	5	4	3	2	1	0
0	0	FILL		BORDER		OUTLINE	
X'004C CROSS-HAIR PATTERN COLOR REGISTER							

7	6	5	4	3	2	1	0
0	Horizontal Pattern						
0	Vertical Pattern						
X'004D HORIZONTAL CROSS-HAIR PATTERN REGISTER ***X'004E VERTICAL CROSS-HAIR PATTERN REGISTER***							

X'005F DAC Control Register

7	6	5	4	3	2	1	0
0	0	0	0	DAC	10B	SEN	SRC

DAC	***DAC Current Control*** *0* Normal Operation *1* No DAC output current	The DAC outputs should be disabled in VIDEO mode to save power. DAC output recovery to normal operation takes 2 ms.
10B	***10b DAC Select*** *0* 9b Mode *1* 10b Mode	In 9b mode the DAC LSB is forced to 0, all other data operations remain at 10 bits.
SEN	***DAC Shunt Enable*** *0* Shunt Disabled *1* Shunt Enabled	DAC Shunt Enable connects the complimentary DAC outputs to an internal analog ground. This is a test feature and is not recommended as the normal DAC termination method.
SRC	***DAC Slew Rate Control*** *0* Fast 2.5 ns *1* Slow 7.5 ns	A rise time selection inconsistent with monitor performance requirements will result in poor image quality. Older low-performance monitors may not be able to accept fast DAC slew rates without generating excessive EMI or RFI noise. Selecting the slow slew rate may minimize such problems.

Diagnostic Registers

TESTABILITY and DIAGNOSTICS

MISR Registers

A Multiple Input Shift Register (MISR) is used to enhance testability of the VRAM to RGB561 interface. The MISR continually processes the 30 bit digital DAC pixel input for a frame of data into a signature for that data and stores it in 4 eight bit registers that make up the MISR (Table 21). This signature is then read from and compared with the correct signature for the specific frame of data used, to determine if a fault exists. By using the VRAM Mask Registers to block specific pixel inputs and using MISR diagnostics, the fault can be isolated to section of circuitry, a card net or VRAM module.

The MISR is enabled by register ***CONF/3*** (X'0003) MISR bit. After enabling, the MISR is reset to X'3FFFFFFF at the next active vertical blanking time and begins accumulating a frame signature when vertical blanking becomes inactive. The MISR enable bit is not reset automatically. To collect another signature the MISR must be disabled for at least 1 frame, then re-enabled.

The signature stored in the MISR registers can be accessed though the MPI at any time, however, the signature data is inverted when read out on the **$DATA_{7-0}$** bus. The 30 bit signature data locations within the MISR registers are shown below. In interlaced mode (CONF/3, IC) the MISR starts accumulating a frame signature on the first even field, continues through the odd field and stops.

X'0060-0063 MISR Signature Registers

Table 21. MISR Register Bit Locations

ADDR	7	6	5	4	3	2	1	0
X'0060 MISR 0	7	6	5	4	3	2	1	0
X'0061 MISR 1	15	14	13	12	11	10	9	8
X'0062 MISR 2	23	22	21	20	19	18	17	16
X'0063 MISR 3	—	—	29	28	27	26	25	24

X'0065 MISR Status

The ***MISR Status Register*** (X'0065) has been provided to assist in monitoring this diagnostic operation. The MISR STATUS bit indicates the MISR register status: reset, active or finished.

It is recommended that the screen be blanked while the MISR operation runs test frames by disabling the DAC outputs, CONF/2 (X'0002), SCR.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	STATUS	
X'0065 MISR STATUS REGISTER (Read)							

STATUS	***MISR Status***	
	00	MISR is reset
	01	Active collecting a frame signature
	10	Finished the MISR operation
	11	Invalid

MISR Algorithm

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

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

MISR Example

The input data used in this example has a value of X'00000200 for each pixel clock cycle, the MISR register reset value is X'3FFFFFFF and the signature is compiled for 2 pixel cycles. The MISR bit 0 (LSB) is on the right, bit 29 (MSB) the left.

A third cycle would yield the value X'3C7FF1ED. The MISR registers and corresponding signature bit locations are given in the table. The values specified are those stored in the MISR registers, when accessed through the MPI **DATA**$_{7-0}$ bus, the bits are inverted.

```
00 0000 0000 0000 0000 0010 0000 0000   Pixel data
11 1111 1111 1111 1111 1111 1111 111    Shifted MISR data
       1                          111   Bit 29
-------------------------------------
11 1111 0111 1111 1111 1101 1111 1001   Cycle 1
                                        X'3F7FFDF9

00 0000 0000 0000 0000 0010 0000 0000   Pixel data
11 1110 1111 1111 1111 1011 1111 001    MISR data shifted
       1                          111   Bit 29
-------------------------------------
11 1110 0111 1111 1111 1001 1111 0101   Cycle 2
                                        X'3E7FF9F5
```

X'0050-0056 VRAM Bit Mask Registers

The VRAM Mask Registers are enabled by VM, CONF/2 (X'0002) and are used to mask (set to 0) a corresponding 4 bit group of VRAM PIX data, for the purpose of isolating defective VRAM modules, card nets or chip circuitry, when used in conjunction with MISR diagnostics. If an incorrect MISR frame signature is obtained with masking disabled, additional fault isolation can be achieved by collecting new signatures with selected VRAM inputs masked and comparing them with correct signatures. The register bits and the corresponding pixel data bits masked are shown below.

X'0064 DAC Comparator

The DAC Comparator is used to verify DAC output levels. Comparisons are made during active screen time using stable DAC output levels of 2 µs duration and the 0.35 V CVREF reference input. Results are latched on the falling edge of vertical blanking and may be read from the register during blanking on the I/O DATA$_{7-0}$ port. This technique can be used to detect DAC faults and the presence and the type of monitor (color/monochrome) being used.

7	6	5	4	3	2	1	0
0	0	0	0	0	BCR	GCR	RCR
X'0064 DAC COMPARATOR REGISTER (Read Only)							

BCR	***Blue DAC Compare Result***
	0 Blue > VREF (input X'269-3FF)
	1 Blue < VREF (input X'000-19A)
GCR	***Green DAC Compare Result***
	0 Green > VREF (input X'269-3FF)
	1 Green < VREF (input X'000-19A)
RCR	***Red DAC Compare Result***
	0 Red > VREF (input X'269-3FF)
	1 Red < VREF (input X'000-19A)

ADDR	7	6	5	4	3	2	1	0
X'0050 BMR 1	31 - 28	27 - 24	23 - 20	19 - 16	15 - 12	11 - 8	7 - 4	3 - 0
X'0051 BMR 2	63 - 60	59 - 56	55 - 52	51 - 48	47 - 44	43 - 40	39 - 36	35 - 32
X'0052 BMR 3	95 - 92	91 - 88	87 - 84	83 - 80	79 - 76	75 - 72	71 - 68	67 - 64
X'0053 BMR 4	127-124	123-120	119-116	115-112	111-108	107-104	103-100	99 - 96
X'0054 BMR 5	159-156	155-152	151-148	147-144	143-140	139-136	135-132	131-128
X'0055 BMR 6	191-188	187-184	183-180	179-176	175-172	171-168	167-164	163-160
X'0056 BMR 7	—	—	—	—	—	—	199-196	195-192

DDOTCLK Register

X'0082 Divided DOT Clock Control Register

7	6	5	4	3	2	1	0
DOT	DOTS			0	0	0	0

DOT	***DOTCLK Enable*** ***0*** Disabled - tristate ***1*** Enable DDOTCLK output	DDOTCLK is a timed output used to sample $VIDEO_{11-0}$ data as shown in Figure 8.
DOTS	***DOT_CLK Output Signal Select*** ***000*** SERIAL_CLK ***001*** PIXCLK ***010*** PIXCLK÷2 ***011*** PIXCLK÷4 ***100*** PIXCLK÷8 ***101*** PIXCLK÷16 ***110*** INVALID ***111*** INVALID	DDOTCLK is a multipurpose timing reference output. It can be programmed as any of the valid options shown. PIXCLK is either the external reference (EXTCLK) or the PLL driven pixel clock. For use with the $VIDEO_{11-0}$ data port DOTS should be set to 001, this will provide a clock edge with each VIDEO data value.
BIT 3	***Reserved***	Reserved bit. Must be set to 0 for proper operation.
BIT 2	***Reserved***	Reserved bit. Must be set to 0 for proper operation.
BIT 1	***Reserved***	Reserved bit. Must be set to 0 for proper operation.
BIT 0	***Reserved***	Reserved bit. Must be set to 0 for proper operation.

Cursor Look-Up Table

X'0A10-0A1F Cursor/Cross-Hair Look-Up Tables

Table 22. Cursor/Cross-hair look-up table entries

ADDRESS	FUNCTION
X'0A10	Transparent
X'0A11	CURSOR PRIMARY COLOR 1
X'0A12	CURSOR PRIMARY COLOR 2
X'0A13	CURSOR PRIMARY COLOR 3
X'0A14	Transparent
X'0A15	CURSOR BLINK COLOR 1
X'0A16	CURSOR BLINK COLOR 2
X'0A17	CURSOR BLINK COLOR 3
X'0A18	Transparent
X'0A19	CROSS-HAIR PRIMARY COLOR 1
X'0A1A	CROSS-HAIR PRIMARY COLOR 2
X'0A1B	CROSS-HAIR PRIMARY COLOR 3
X'0A1C	Transparent
X'0A1D	CROSS-HAIR BLINK COLOR 1
X'0A1E	CROSS-HAIR BLINK COLOR 2
X'0A1F	CROSS-HAIR BLINK COLOR 3

Window Attribute Tables

X'0E00-0E0F AUX_FB_WAT

7	6	5	4	3	2	1	0
0	0	0	0	0	GMA	XH	PT

GMA	***GAMMA LUT Bypass Enable*** *0* Use GAMMA LUTs *1* Bypass GAMMA LUTs	Indicates if GAMMA correction should be applied to the pixel color LUT data or if it should go directly to the DAC inputs.
XH	***Cross-Hair Enable*** *0* No Cross-Hair *1* Enable Cross-Hair	XH enables the cross-hair cursor for the frame buffer pixel layer. This is valid only if WID clipping is enabled in the ***Cross-Hair Control Register*** (X'0031).
PT	***Pixel Transparency Value*** *0* X'00 - transparent *1* X'FF - transparent	If pixel data matches the specified value and transparency is enabled in the FB_WAT, the frame buffer pixel data is not displayed.

X'0F00-0F0F AUX_OL_WAT

7	6	5	4	3	2	1	0
0	0	CK	UL	OL	GB	XH	OT

CK/OT	***Chroma Key/OL Transparency*** *00* X'00 *01* X'FF *10* Chroma Key 0/Mask 0 Reg *11* Chroma Key 1/Mask 1 Reg	CK/OT selects the value to be compared with overlay data to determine transparency. Overlay data will be transparent if Overlay Transparency is enabled in the OL_WAT and the overlay data matches either X'00 or FF or the transparency value stored in the ***Chroma Key Registers*** (X'0010-0011), comparing only those bits selected in the ***Chroma Key Mask Registers*** (X'0012-0013).
UL	***Underlay Enable*** *0* Disabled *1* Enabled	Underlay is considered disabled for OL_WAT pixel formats PIX_FORM=00, 11.
OL	***Overlay Enable*** *0* Disabled *1* Enabled	Enables overlay.
GB	***OL/UL GAMMA LUT Bypass*** *0* Use GAMMA LUTs *1* Bypass GAMMA LUTs	Indicates if GAMMA correction should be applied to the OL/UL data or if it should go directly to the DAC inputs.
XH	***Cross-Hair Enable*** *0* No Cross-Hair *1* Enable Cross-Hair	XH enables the cross-hair cursor for the overlay layer. This is valid if overlay data is displayed. The cross-hair is considered disabled if underlay is to be displayed. This option is valid if WID clipping is enabled in the Cross-Hair Control Register (X'0031).

X'1000-10FF FB_WAT

9	8	7	6	5	4	3	2	1	0
START_ADDR				PIXFORM		BS	MODE		TR

START_ ADDR	***Color LUT Start Address***	The START_ADDR bits are added to the FB_WID bits to generate the Color LUT address as shown in Table 1 on page 2. The start address allows individual 64 entry Color LUTs.
PIX_ FORM	***FB Pixel Format*** ***00*** 8 bpp ***01*** 12 bpp ***10*** 16 bpp ***11*** 24/30 bpp	Bit-per-pixel format options for VRAM input data programmed in this register should be consistent with those valid options listed in Table 7 on page 9.
BS	***Buffer Select*** ***0*** Frame Buffer A ***1*** Frame Buffer B	Buffer Select specifies the frame buffer field, FB_A or FB_B, of the VRAM input data from which to select pixels.
MODE	***Color Mode*** ***00*** Index ***01*** Grey Scale ***10*** Direct RGB ***11*** True	Grey Scale and True color modes bypass the Color Look-up Tables and present the pixel data directly to the DAC inputs. Refer to Figure 5 on page 19 for a mode description.
TR	***Transparency Enable*** ***0*** Opaque ***1*** Pixel Transparency	An opaque pixel blocks underlay data, a transparent pixel displays underlay data. If there is no underlay data, pixel data is displayed. The transparency value for the FB is stored in the AUX_FB_WAT.

X'1400-14FF OL_WAT

9	8	7	6	5	4	3	2	1	0
START_ADDR				PIXFORM		BS	MODE		TR

START_ ADDR	***Color LUT Start Address***	The START_ADDR bits are added to the OL_WID bits to generate the Color LUT address as shown in Table 1 on page 2. The start address allows individual 64 entry Color LUTs.
PIX_ FORM	***Overlay Pixel Format*** ***00*** 8b OL / 0b UL ***01*** 6b OL / 2b UL ***10*** 4b OL / 4b UL ***11*** 4b OL - Double Buffered	Underlay is disabled for 00, 11 selections regardless of the value of the Underlay Enable bit (UL) in the AUX_OL_WAT.
BS	***Buffer Select for Overlay*** ***0*** Frame Buffer A ***1*** Frame Buffer B	Refer to Table 7 on page 9 for the frame buffer data locations corresponding to the various overlay formats
MODE	***Color Mode for Overlay*** ***00*** Index ***01*** Grey Scale ***10*** Indirect ***11*** Direct	Grey Scale and True color modes bypass the Color Look-up Tables and present the overlay data directly to the DAC inputs. Refer to Figure 5 on page 19 for a mode description.
TR	***Transparency Enable*** ***0*** Opaque ***1*** Overlay Transparency	Overlay transparency is determined when it matches the transparency value specified in the AUX_OL_WAT: X'00, FF or an arbitrary value stored in the Chroma Key Registers. The options are set on a pixel basis by the ***AUX_OL_WAT*** CK/OT bits.

Cursor Icon Pixel Map

X'2000-23FF Cursor Pixel Map

The 64 x 64 cursor is mapped into a 1K x 8 SRAM as indicated in Tables 3, and 4. Each 8b input to the cursor pixmap represents 4 pixel color values. The 2b/pixel data is used to access the Cursor LUT to select a primary or blink color based on the value in the ***Cursor Control Register*** (X'0030), BT.

Table 23. 64x64 Cursor Pixel Screen Locations

Addr		0	1	2	3	...	60	61	62	63
X'2000	*0*	0	1	2	3	...	60	61	62	63
X'2010	*1*	64	65	66	67	...	124	125	126	127
X'2020	*2*	128	129	130	131	...	188	189	190	191
X'2030	*3*	192	193	194	195	...	252	253	254	255
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
X'23C0	*60*	3840	3841	3842	3843	...	3900	3901	3902	3903
X'23D0	*61*	3904	3905	3906	3907	...	3964	3965	3966	3967
X'23E0	*62*	3968	3969	3970	3971	...	4028	4029	4030	4031
X'23F0	*63*	4032	4033	4034	4035	...	4092	4093	4094	4095

Addr is the register address location of the first 4 pixels Register bit placement is described in Table 4.

Table 24. Cursor Pixels Mapped into 1Kx8 Memory

ADDRESS	DATA$_{7-0}$ Bits							
	7	6	5	4	3	2	1	0
X'2000	Pixel 0		Pixel 1		Pixel 2		Pixel 3	
	Bit 1	Bit 0	Bit 1	Bit 0	Bit 1	Bit 0	Bit 1	Bit 0
X'2001	Pixel 4		Pixel 5		Pixel 6		Pixel 7	
	Bit 1	Bit 0	Bit 1	Bit 0	Bit 1	Bit 0	Bit 1	Bit 0
⋮								
X'23FE	Pixel 4088		Pixel 4089		Pixel 4090		Pixel 4091	
	Bit 1	Bit 0	Bit 1	Bit 0	Bit 1	Bit 0	Bit 1	Bit 0
X'23FF	Pixel 4092		Pixel 4093		Pixel 4094		Pixel 4095	
	Bit 1	Bit 0	Bit 1	Bit 0	Bit 1	Bit 0	Bit 1	Bit 0

Signal Pins

Table 25 (Page 1 of 3). SIGNAL PINS				
SIGNAL	**#**	**I/O**	**NAME**	**DESCRIPTION**
DATA PORT (201)				
$PIX_{199\text{-}0}$	200	INPUT	VRAM DATA	The pixel, overlay and window ID data from the frame buffer can be segmented in various ways set by the CONF/1 register and summarized in Table 7 on page 9. The pixel input data from the VRAM frame buffer may also be selectively masked by the ***VRAM MASK Registers*** (X'0050-0056) for diagnostics. Unused inputs should be terminated (10KΩ to V_{DD}).
LOAD_CLK	1	INPUT	LOAD CLOCK	The LOAD_CLK gates pixel input data from the VRAM frame buffer into the RGB561 serializer. It is equivalent to the VRAM serial clock and may be provided by the controller or the RGB561.
MPI and VIDEO SIGNALS (16)				
$DATA_{7\text{-}0}$/ $VIDEO_{7\text{-}0}$	8	BIDI	DATA BUS/ VIDEO BUS LSBs	$DATA_{7\text{-}0}$ is the MPI data bus used to read and write internal control registers which set the operating mode. It is also used to load the color palettes, gamma tables and cursor pixel map and read diagnostic registers. It is controlled by $\overline{CE}$, R/W, C0 and C1 with the timings of Figure 6. The $VIDEO_{7\text{-}0}$ bus provides digital video output data prior to conversion by the DACs into an RGB signal for external use. In VIDEO mode, the MPI $DATA_{7\text{-}0}$ is available during blanking.
$VIDEO_{11\text{-}8}$	4	OUTPUT	VIDEO BUS MSBs	These are the remaining 4 bits of the 12 bit digital VIDEO bus. Refer to Figure 8 for timings and "Video" on page 5 for additional details.
$\overline{CE}$	1	INPUT	CHIP ENABLE	Chip Enable controls the read/write operations of MPI $DATA_{7\text{-}0}$ to selected register addresses. Timings are shown in Figure 6.
C0, C1	2	INPUT	COMMAND LINE 0/1	The command signals are used with $\overline{CE}$ to control loading and incrementing the address index register. Table 20 lists valid register addresses and their appropriate command setting. ***C1 C0 ACTION*** **0 0** Access the Address Index Register - low byte **0 1** Access the Address Index Register - high byte **1 0** Access the 8b location specified in the Address Index register **1 1** Access the LUT/WAT location in the Address Index register
R/W	1	INPUT	READ / WRITE	Read=1, Write=0 Used with $\overline{CE}$ to access register locations.
DAC OUTPUTS (6)				
RED, GREEN, BLUE	3	OUTPUT ANALOG	DAC RGB OUTPUTS	These analog monitor outputs are RS-343A compatible, have internal clamping and flash-over protection and are capable of driving doubly terminated 75 or 100 Ω coax without buffering or external components.
$\overline{RED}$, $\overline{GREEN}$, $\overline{BLUE}$	3	OUTPUT ANALOG	COMPLEMENTARY DAC RGB OUTPUTS	These are the complement signals to the Red, Green, Blue outputs. When not used, they should be terminated to analog ground.
DTG SIGNALS (9)				
HSYNC	1	OUTPUT	HORIZONTAL SYNC	HSYNC is provided by the Display Timing Generator, and is enabled with signal characteristics set in the ***DTG Control register*** (X'0020). HSYNC is tri-stated for SYNC-on-GREEN (SOG) but can be reprogrammed as an output. External buffering is required for loads greater than 25 pf. When not used it should be terminated (10KΩ to V_{DD}).
CSYNC	1	INPUT	COMPOSITE SYNC	Composite SYNC input from the controller.
$\overline{CBLANKIN}$	1	INPUT	COMPOSITE BLANK	Composite blank (CBLANK) input received from a controller.

Table 25 (Page 2 of 3). SIGNAL PINS

SIGNAL	#	I/O	NAME	DESCRIPTION
$\overline{\text{CBLANKOUT}}$	1	OUTPUT	COMPOSITE BLANK	This is a composite blank output signal, synchronized to the digital $\text{VIDEO}_{11\text{-}0}$ outputs, indicating when the outputs are valid or the display is in blanking mode. Timings are shown in Figure 8.
SERIAL_CLK	1	OUTPUT	VRAM SERIAL CLOCK	The SERIAL_CLK is used to transfer pixel data from the VRAM serial port to the RGB561 frame buffer interface. If this signal is provided by a controller to the VRAMs and the RGB561 this output should be tri-stated and terminated (10KΩ to V_{DD}). If desired the RGB561 SERIAL_CLK output can be enabled as a free running serial clock based on the multiplex mode setting in the ***Configuration 1 Register*** (X'0001), MUX. The far end of the signal is used as the LOAD_CLK input. External buffering is required if the SERIAL_CLK load exceeds 25 pf.
AUX_ SERIAL_CLK	1	OUTPUT	AUXILIARY SERIAL CLOCK	Programmable early SERIAL_CLK. Can be 0 to 7 clock periods earlier than the SERIAL_CLK output as specified by the ***AUX_SERIAL_CLK Control Register*** (X'0023).. It can be used as a second SERIAL_CLK for loading or timing reasons, or it can used in conjunction with the RGB061 to control VRAMs in large double buffered configurations to avoid dotting on the RGB561 serializer. When not used it should be terminated (10KΩ to V_{DD}).
DDOTCLK	1	OUTPUT	DIVIDED DOT CLOCK	The DDOTCLK is a programmed timing reference for the VIDEO outputs which can be either a divided internal pixel clock, driven from the PLL or EXTCLK, or a delayed SERIAL_CLK. It is programmed in the ***DTG Timing Reference Register*** X'0082, DOTS. Timings are shown in Figure 8.
FIELD	1	INPUT	FIELD INPUT	The FIELD signal is used with interlaced monitor operation and indicates when an even or odd scan line of data is being displayed. It is used to control interleave and cursor data generation. The even/odd polarity can be programmed in the CONF/3 register. It is sampled on the rising edge of $\overline{\text{CBLANKIN}}$.
$\overline{\text{RESET}}$	1	INPUT	REGISTER RESET	$\overline{\text{RESET}}$ sets all registers to X'00 and tri-states all outputs while active. A minimum 1 μs pulse width is required to reset registers. The color, gamma and cursor tables, cursor pixel map, revision and reserved registers are unaffected.
TEST SIGNALS (4)				
$\overline{\text{RI}}$	1	INPUT	RECEIVER INHIBIT	$\overline{\text{RI}}$ disables all receiver inputs during module testing and should be terminated (10KΩ to V_{DD}) if not used.
$\overline{\text{DI1}}$, $\overline{\text{DI2}}$	2	INPUT	DRIVER INHIBIT	These signals tri-state all output drivers during module testing and should be terminated (10KΩ to V_{DD}) if not used.
TEST	1	INPUT	TEST ENABLE	Enables module test mode and activates the other test signals. No termination is required.
EXTERNAL COMPONENTS / CIRCUITRY (9)				
REFCLK	1	INPUT	LOW FREQUENCY REFERENCE	This input is used as the PLL reference frequency and can be any TTL or CMOS oscillator frequency from 4 to 100 MHz. If the PLL is not used, this input must be terminated (10KΩ to V_{DD}).
EXTCLK / $\overline{\text{EXTCLK}}$	2	INPUT	EXTERNAL ECL PIXEL CLOCK	When the PLL is not used, a differential ECL oscillator must provide the pixel clock at a frequency consistent with monitor pixel data rates on these inputs. If the module PLL is used, these inputs require termination (EXTCLK - 10KΩ to V_{DD}, $\overline{\text{EXTCLK}}$ - 75Ω to GND).
VREF, GREF, CVREF, RREF	4	DAC INPUTS	Current Reference Gate Reference Comparator Reference Gain Control	External 1.235 V reference used to set full scale currents Gate current source reference A .35 V reference for the DAC comparators An op-amp component compensating current temp/voltage variations See "Circuit Schematic" on page 54 for the required external component connections to these inputs.

Table 25 (Page 3 of 3). SIGNAL PINS				
SIGNAL	**#**	**I/O**	**NAME**	**DESCRIPTION**
PLLCAP, PLLCAP RET	2	PLL INPUTS	External filter connection External filter return	External VCO filter component connections See "Circuit Schematic" on page 54 for the external component connections to these pins.
POWER SUPPLIES (59)				
VDD AVDD P3_PLL	16 6 1	SUPPLY	DIGITAL (3.1 - 3.46 V) DAC (3.1 - 3.46 V) PLL (3.1 - 3.46 V)	Separate digital/analog voltage planes are required. Decoupling from digital supply should be done with a 1nH inductor or ferrite bead connected at one point.
GND AGND P0_PLL	23 6 1	SUPPLY	DIGITAL GROUND DAC GROUND PLL GROUND	Separate digital/analog ground planes are not recommended. For details on supply decoupling refer to "Analog Voltages" on page 53
No Connect	6	—	—	No termination is allowed.

External Circuitry

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

Analog Voltages

The DAC and PLL require external components be attached as shown in "Circuit Schematic" on page 54 and summarized in "Component Values" on page 54. All components are connected to ANALOG supplies, AV_{DD} or AGND.

The card GND plane should be solid.

The card VDD plane should be segmented with separate PLL and DAC supplies. These can be derived from the digital supply through a 1 nH inductor, with separate bulk and high frequency decoupling capacitors. Care should be taken to ensure digital signals do not radiate noise into quiet analog circuits and should not be wired over the analog power planes AVDD/P3PLL.

The on chip PLL is very sensitive to on card generated noise, for optimal performance please refer to the RGB Palette DAC Card Design Guidelines for additional information on reducing noise in high performance card designs.

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

DAC Outputs

Ideally, RGB signals should be wired on a separate signal planes with adjacent AGND wires running in parallel on each side to isolate them from potential digital signal noise generation.

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

PLL Components

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

Signal Terminations

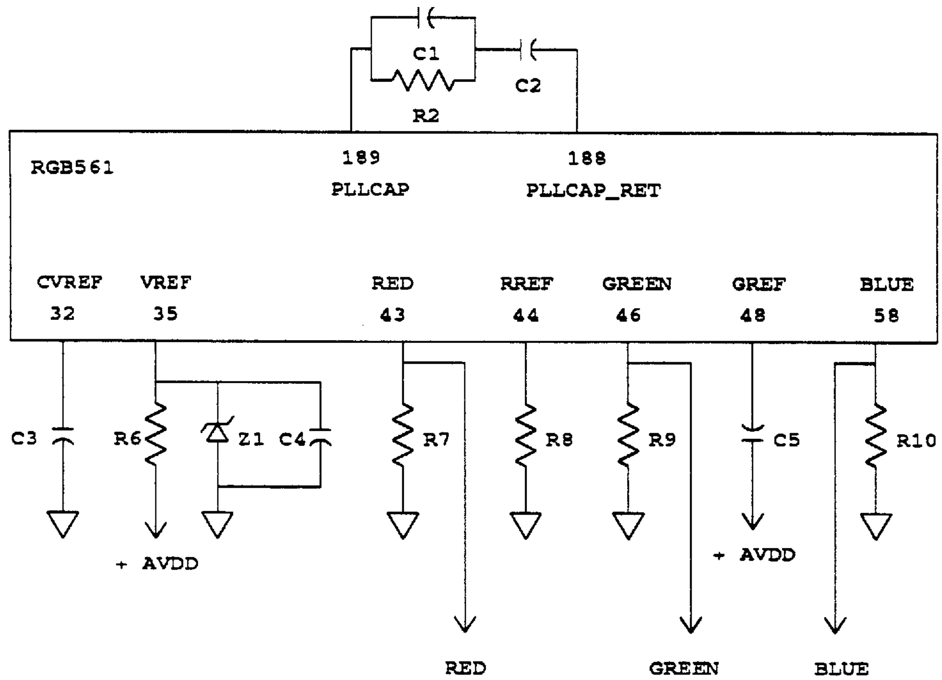
Signals that are either unused inputs or tri-stated outputs require termination. Refer to Table 25 on page 50 to determine if an I/O is used in the mode selected or if it should be terminated.

Pin	Name	Status	Termination
52	HSYNC	Tri-state	10 KΩ to V_{DD}
All 34 182 196 198 211 214 217	PIX_{199-0} DI2 DI1 RI FIELD REFCLK SERIAL_CLK AUX_SERIAL_CLK	Unused	10 KΩ to V_{DD}
203 204	EXTCLK $\overline{\text{EXTCLK}}$	Unused	10 KΩ to V_{DD} 75 Ω to GND

Decoupling

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

Circuit Schematic



Component Values

Component	Value	Tol	Vendor P/N
R2	1.3 KΩ	5%	PANASONIC ERJ3GVYJ132S
R6	1.0 KΩ	5%	PANASONIC ERJ3GVYJ102S
R7, R9, R10	75 Ω, 100 Ω	1%	match video cable impedance
R8	704 Ω	1%	for doubly terminated 75 Ω DAC output
	938 Ω	1%	for doubly terminated 100 Ω DAC output
C1	680 pF	10%	KYOCERA 1206C681K3B05
	680 pF	5%	VITRAMON
C2	8.2 nF	10%	–
C3, C5	0.001 μF	10%	KYOCERA 0603X102K2B02
C4	0.01 μF	10%	KYOCERA 1206X103K2B02
Z1	1.2V REF		NATIONAL SEMICONDUCTOR LM385-1.2

▪AC Characteristics

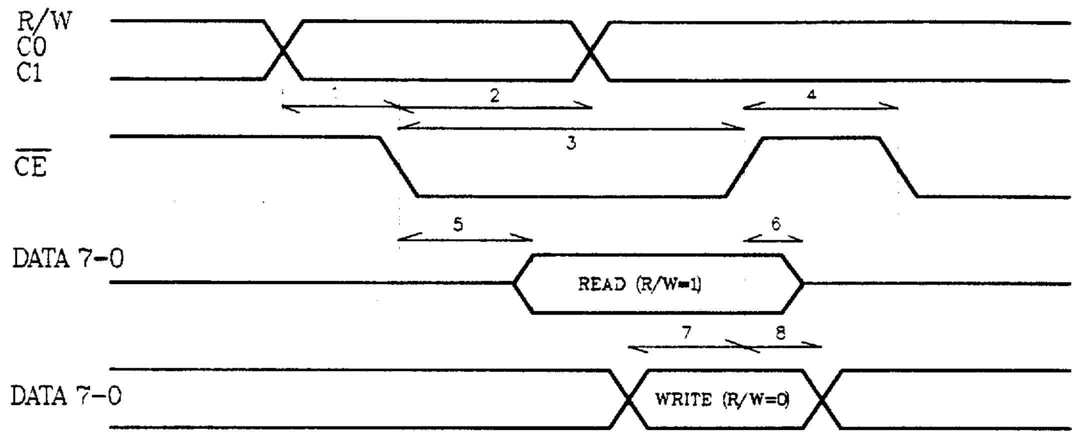


Figure 6. MPI Port Timings

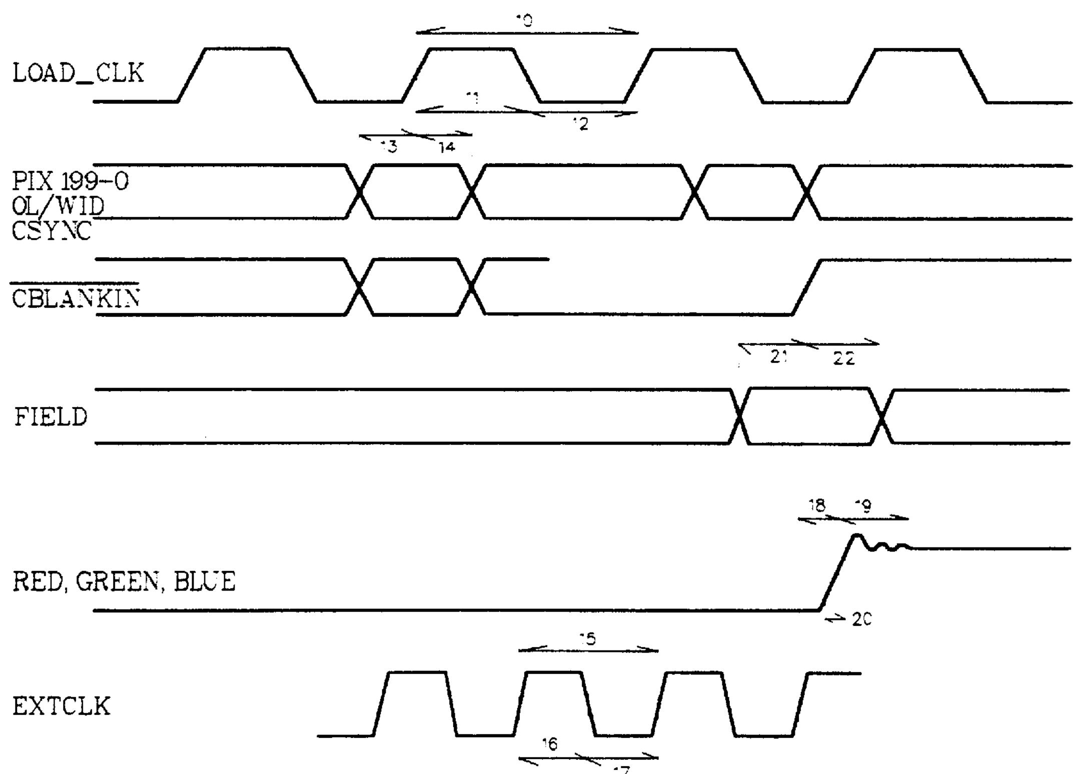


Figure 7. VRAM Pixel Port and Clock Timings

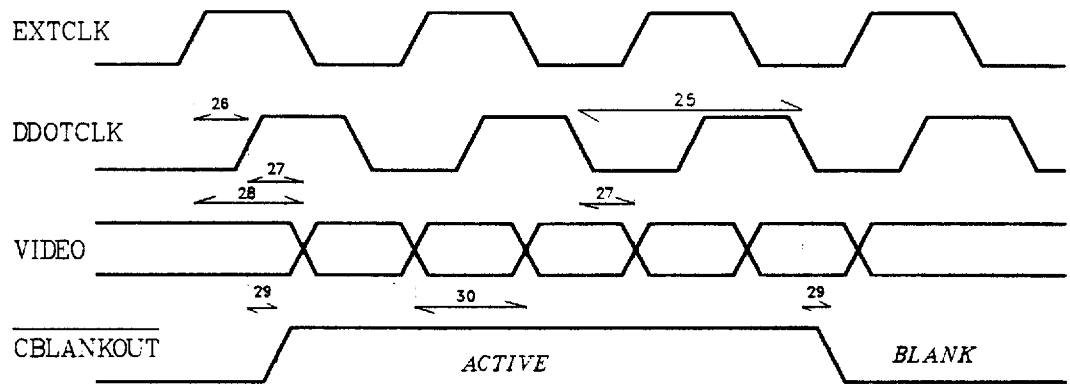


Figure 8. VIDEO Timings. The DDOTCLK output is programmed by the Divided DOT Clock Control Register, X'0082.

Parameter	Symbol	250 MHz		170 MHz		Units
		min	max	min	max	
Pixel clock (EXTCLK) rate	f_{MAX}		250		170	MHz
LOAD_CLK rate	LD_{MAX}					
4:1 multiplexing			62.5		42.5	MHz
5:1 multiplexing			50.0		34.0	MHz
8:1 multiplexing			32.3		21.3	MHz
R/W, C0, C1 Controls						
Setup	1	10		10		ns
Hold	2	5		5		ns
$\overline{CE}$ Timings for C0/C1 ≠ 1/1 or Addresses Index below X'0A00.						
$\overline{CE}$ Signal Parameters						
Low	3	50		50		ns
High	4	25		25		ns
To Data Valid	5		25		25	ns
To Data Bus Tri-state	6	2		2		ns
$\overline{CE}$ Timings for C0/C1 = 1/1 or Addresses Index above X'0A00. PC = Pixel clock (PIXCLK) Cycle in ns						
$\overline{CE}$ Signal Parameters						
Low (Read)	3	4 PC		4 PC		ns
Low (Write)	3	3 PC		3 PC		ns
High (the greater of)	4	2.5 PC or 25		2.5 PC or 25		ns
To Data Valid	5	25	4 PC + 15	25	4 PC + 15	ns
To Data Bus Tri-state	6	2		2		ns
$DATA_{7-0}$ Valid (Write)						
Setup	7	10		10		ns
Hold	8	5		5		ns
LOAD_CLK Cycle Time	10					
4:1 multiplexing		16		23.5		ns
5:1 multiplexing		20		29.4		ns
8:1 multiplexing		32		47.1		ns
LOAD_CLK Pulse Width	11/12	1.5 PC		1.5 PC		ns
$LOAD\_CLK = \left(\frac{1}{f_{max}}\right) MUX\ (ns)$						
VRAM Data and Control Inputs						
Setup	13	3		3		ns
Hold	14	2		2		ns
PIXCLK Signal						
Cycle time	15	4.0		5.9		ns
Pulse Width High	16	1.6		2.5		ns
Pulse Width Low	17	1.6		2.5		ns
DAC Analog Outputs						
Output delay	18		3		6	ns
Settling time	19		2		5	ns
Rise/fall	20		1		10	ns
FIELD						
Setup	21	10		10		ns
Hold	22	5		5		ns

Parameter	Symbol	100 MHz max		Units
		min	max	
VIDEO Mode Timings				
DDOTCLK period	25	10	-	ns
EXTCLK to DDOTCLK Delay	26	-	-	ns
DDOTCLK to VIDEO Access	27		7	ns
EXTCLK to VIDEO Access	28	-	-	ns
DDOTCLK to $\overline{\text{CBLANKOUT}}$	29		7	ns
VIDEO Pulse Width	30	4	-	ns

9006146 0000277 219

▪DAC Output Levels

RREF = 938 OHMS				RREF = 704 OHMS			
RED, BLUE		GREEN		RED, BLUE		GREEN	
mA	V	mA	V	mA	V	mA	V
13.99	0.698	20.00	1.00	18.65	0.698	26.67	1.00
0.00	0.000	8.01	0.302	0.00	0.000	8.05	0.302
		0.00	0.000			0.00	0.000

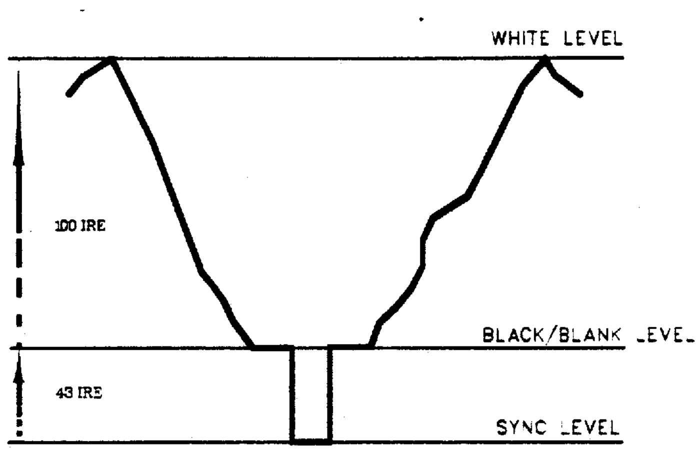


Figure 9. Composite DAC Output (Setup=0 IRE). For 100Ω and 75Ω doubly terminated loads, RS-343A levels, Blank Pedestal=0 IRE, SYNC on Green.

RREF = 938 OHMS				RREF = 704 OHMS			
RED, BLUE		GREEN		RED, BLUE		GREEN	
mA	V	mA	V	mA	V	mA	V
14.28	0.714	20.00	1.000	19.05	0.714	26.67	1.000
1.04	0.054	6.78	0.340	1.44	0.054	9.05	0.340
0.00	0.000	5.71	0.286	0.00	0.000	7.62	0.286
		0.00	0.000			0.00	0.000

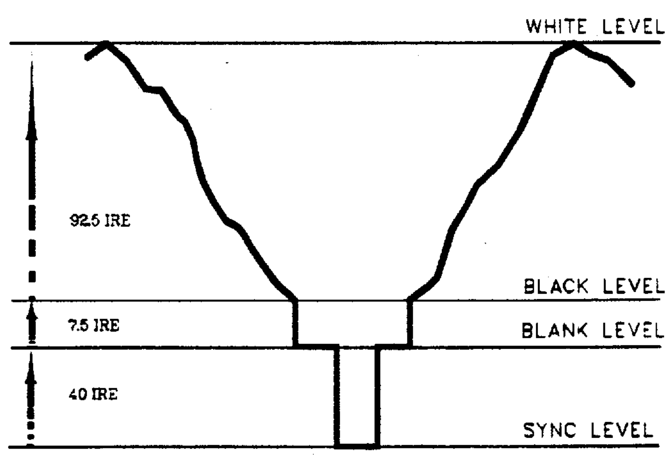


Figure 10. Composite DAC Output (Setup=7.5 IRE). For 100Ω and 75Ω doubly terminated loads, RS-343A levels, Blank Pedestal=7.5 IRE, SYNC on Green.

▪DC Characteristics

Item	Symbol	Conditions	MIN	TYP	MAX	Units
Digital Inputs						
Input Levels (TTL) High	V_{IH}		2.0	5.0	5.25	V
Low	V_{IL}		0.0	0	0.8	
ECL Levels - High	V_{EIH}		2.3		2.95	V
Low	V_{EIL}		1.5		2.1	
Common Mode	V_{CM}		2.0	2.2	2.4	
Differential	V_{Δ}		0.30		1.0	
Differential Input Current	I_{IH}	$V_{IN}=3.3V$	0		1	μA
	I_{IL}	$V_{IN}=-0.5V$	0		-1	
Input Capacitance	C_I	f=1 MHz			10	pF
Digital Outputs						
Output Levels	V_{OH}		2.4	V_{DD}	3.47	V
	V_{OL}		0.0	GND	0.4	
Output Current	I_{OH}	$V_{OH}=2.4V$,	0.0		14.0	mA
	I_{OL}	$V_{OL}=0.4V$	0.0		-7.0	
Output Leakage Current	I_Z	HI-Z	-20		20	μA
Output Impedance	Z_O	Enabled		50		Ω
Output Capacitance	C_O	f=1 MHz			10	pF
DAC Analog Outputs						
Resolution			9	9	10	bits
Accuracy (9 bit)						
Monotonicity		Guaranteed				
Absolute Full Scale	AFS				±5	%
Integral Linearity Error	ILE				±1¼	LSB
Differential Linearity Error	DLE				±½	LSB
Accuracy (10 bit)						
Monotonicity		Typical				
Integral Linearity Error	ILE				±2½	LSB
Differential Linearity Error	DLE				±1	LSB
DAC to DAC Matching					±1	%
Output Transition						
Fast	T_R/T_F				2.5	ns
Slow					7.5	

▪Recommended Operating Conditions

Item	Symbol	Conditions	MIN	TYP	MAX	Units
Power Supply	V_{DD} AVDD P3PLL		3.14	3.3	3.47	V
	GND AGND P0PLL		-0.1 -0.05 -0.05	0	+0.1 +0.05 +0.05	
Operating Supply Current	I_{DD}	V_{DD}=3.47V, f=170 MHz			900	mA
Ambient Temperature	T_A		0		70	°C

▪Absolute Maximum Ratings

Item	Symbol	Rating	Units
Voltage on Any Pin	V_{DD} GND	V_{DD}+0.5 GND-0.5	V
Ambient Temperature	T_A	0 to +85	°C
Module Operating Temperature	T_O	-25 to +125	°C
Storage Temperature	T_S	-55 to +150	°C
Soldering Temperature	T_{SOL}	260	°C
Electrostatic Discharge Protection	ESD	2000	V

Exposing the device to stress in excess of that listed will cause permanent damage.

Power and Cooling

Cooling Requirements

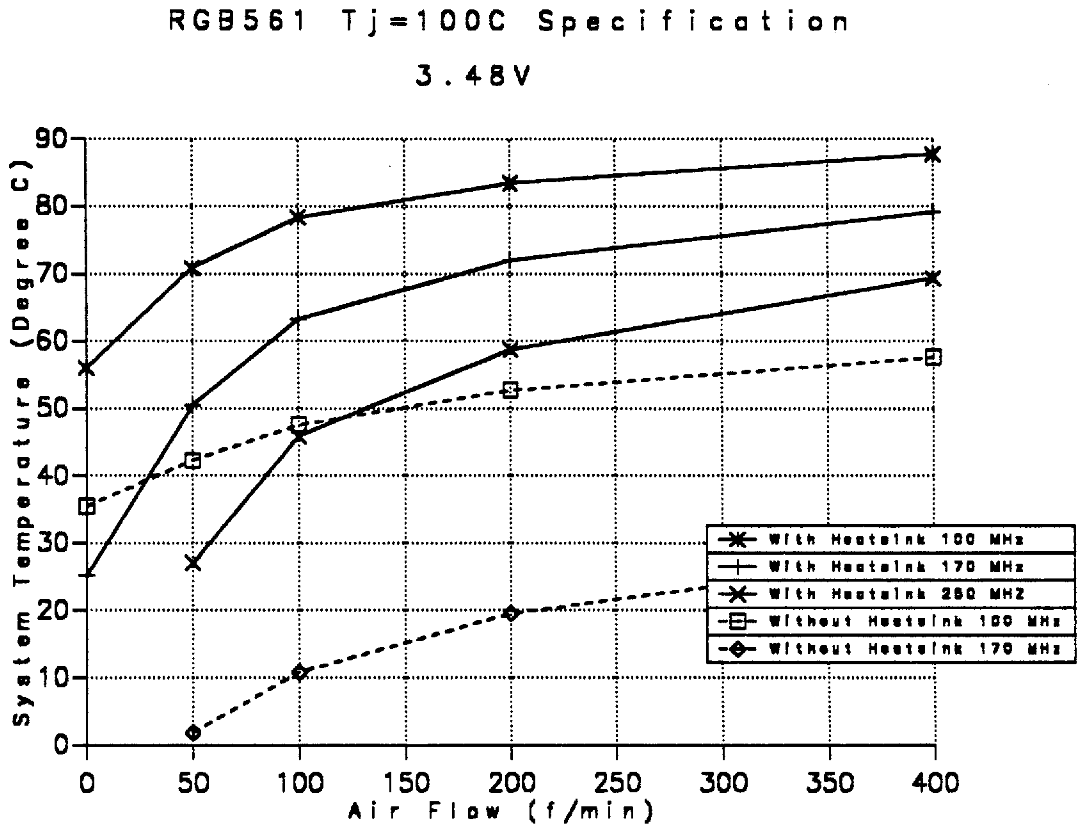


Figure 11. Heatsink cooling example.

Sufficient cooling must be provided to the RGB561 to prevent excessive junction temperatures. For guaranteed reliability to spec. the junction temperature must be maintained at or below 100°C. The chart above shows the effect of ambient air temperature and airflow on cooling effectiveness with a 33x33x15mm, 8x8 pin-fin heatsink. To maintain an acceptable junction temperature at 170 MHz and 3.48V power supply, an airflow of 200 feet per minute would be required for an ambient air temperature (flowing over the heatsink) of 72°C.

Pin Assignments

PIN	SIGNAL
1	GND
2	PIX$_{25}$
3	PIX$_{24}$
4	PIX$_{23}$
5	VDD
6	PIX$_{22}$
7	PIX$_{21}$
8	PIX$_{20}$
9	PIX$_{19}$
10	PIX$_{18}$
11	PIX$_{17}$
12	PIX$_{16}$
13	PIX$_{15}$
14	PIX$_{14}$
15	PIX$_{13}$
16	PIX$_{12}$
17	PIX$_{11}$
18	PIX$_{10}$
19	PIX$_{9}$
20	PIX$_{8}$
21	PIX$_{7}$
22	PIX$_{6}$
23	PIX$_{5}$
24	GND
25	VDD
26	PIX$_{4}$
27	PIX$_{3}$
28	PIX$_{2}$
29	PIX$_{1}$
30	PIX$_{0}$
31	CSYNC
32	CVREF
33	AVDD
34	$\overline{\text{DI2}}$
35	VREF
36	AVDD
37	AGND
38	AVDD
39	GND
40	$\overline{\text{RED}}$
41	AGND
42	AGND
43	RED
44	RREF
45	$\overline{\text{GREEN}}$
46	GREEN
47	TEST
48	GREF
49	AVDD
50	AGND
51	$\overline{\text{CBLANKOUT}}$
52	HSYNC
53	VDD
54	GND
55	AVDD
56	AGND
57	$\overline{\text{BLUE}}$
58	BLUE
59	$\overline{\text{RESET}}$
60	AVDD
61	AGND
62	C1
63	DATA/VIDEO$_{7}$
64	DATA/VIDEO$_{6}$
65	C0
66	DATA/VIDEO$_{5}$
67	DATA/VIDEO$_{4}$
68	DATA/VIDEO$_{3}$
69	DATA/VIDEO$_{2}$
70	DATA/VIDEO$_{1}$
71	$\overline{\text{CE}}$
72	VDD
73	R/W
74	DDOTCLOCK
75	DATA/VIDEO$_{0}$
76	GND
77	GND
78	VIDEO$_{8}$
79	PIX$_{199}$
80	PIX$_{198}$
81	VDD
82	VIDEO$_{9}$
83	PIX$_{197}$
84	VIDEO$_{10}$
85	VIDEO$_{11}$
86	PIX$_{196}$
87	PIX$_{195}$
88	PIX$_{194}$
89	PIX$_{193}$
90	PIX$_{192}$
91	PIX$_{191}$
92	PIX$_{190}$
93	PIX$_{189}$
94	PIX$_{188}$
95	PIX$_{187}$
96	PIX$_{186}$
97	PIX$_{185}$
98	PIX$_{184}$
99	PIX$_{183}$
100	GND
101	VDD
102	PIX$_{182}$
103	PIX$_{181}$
104	PIX$_{180}$
105	PIX$_{179}$
106	PIX$_{178}$
107	PIX$_{177}$
108	PIX$_{176}$
109	PIX$_{175}$
110	PIX$_{174}$
111	PIX$_{173}$
112	PIX$_{172}$
113	PIX$_{171}$
114	PIX$_{170}$
115	GND
116	PIX$_{169}$
117	PIX$_{168}$
118	PIX$_{167}$
119	PIX$_{166}$
120	PIX$_{165}$
121	PIX$_{164}$
122	PIX$_{163}$
123	PIX$_{162}$
124	PIX$_{161}$
125	PIX$_{160}$
126	PIX$_{159}$
127	PIX$_{158}$
128	PIX$_{157}$
129	VDD
130	GND
131	PIX$_{156}$
132	PIX$_{155}$
133	PIX$_{154}$
134	PIX$_{153}$
135	PIX$_{152}$
136	PIX$_{151}$
137	PIX$_{150}$
138	PIX$_{149}$
139	PIX$_{148}$
140	PIX$_{147}$
141	PIX$_{146}$
142	PIX$_{145}$
143	PIX$_{144}$
144	PIX$_{143}$
145	PIX$_{142}$
146	PIX$_{141}$
147	PIX$_{140}$
148	VDD
149	PIX$_{139}$
150	PIX$_{138}$
151	PIX$_{137}$
152	GND
153	GND
154	PIX$_{136}$
155	PIX$_{135}$
156	PIX$_{134}$
157	VDD
158	PIX$_{133}$
159	PIX$_{132}$
160	PIX$_{131}$
161	PIX$_{130}$
162	PIX$_{129}$
163	PIX$_{128}$
164	PIX$_{127}$
165	PIX$_{126}$
166	PIX$_{125}$
167	PIX$_{124}$
168	PIX$_{123}$
169	PIX$_{122}$
170	PIX$_{121}$
171	PIX$_{120}$
172	PIX$_{119}$
173	PIX$_{118}$
174	PIX$_{117}$
175	PIX$_{116}$
176	GND
177	VDD
178	PIX$_{115}$
179	PIX$_{114}$
180	PIX$_{113}$
181	PIX$_{112}$
182	$\overline{\text{DI1}}$
183	GND
184	GND
185	no connect
186	P3_PLL
187	no connect
188	PLLCAP RET
189	PLLCAP
190	no connect
191	P0_PLL
192	no connect
193	no connect
194	GND
195	GND
196	$\overline{\text{RI}}$
197	no connect
198	FIELD
199	PIX$_{111}$
200	PIX$_{110}$
201	PIX$_{109}$
202	PIX$_{108}$
203	EXTCLK
204	$\overline{\text{EXTCLK}}$
205	VDD
206	GND
207	PIX$_{107}$
208	PIX$_{106}$
209	PIX$_{105}$
210	PIX$_{104}$
211	REFCLK
212	PIX$_{103}$
213	PIX$_{102}$
214	SERIAL_CLK
215	PIX$_{101}$
216	PIX$_{100}$
217	AUX_SERIAL_CLK
218	PIX$_{99}$
219	PIX$_{98}$
220	PIX$_{97}$
221	PIX$_{96}$
222	PIX$_{95}$
223	LOAD_CLK
224	VDD
225	PIX$_{94}$
226	$\overline{\text{CBLANKIN}}$
227	PIX$_{93}$
228	GND
229	GND
230	PIX$_{92}$
231	PIX$_{91}$
232	PIX$_{90}$
233	VDD
234	PIX$_{89}$
235	PIX$_{88}$
236	PIX$_{87}$
237	PIX$_{86}$
238	PIX$_{85}$
239	PIX$_{84}$
240	PIX$_{83}$
241	PIX$_{82}$
242	PIX$_{81}$
243	PIX$_{80}$
244	PIX$_{79}$
245	PIX$_{78}$
246	PIX$_{77}$
247	PIX$_{76}$
248	PIX$_{75}$
249	PIX$_{74}$
250	PIX$_{73}$
251	PIX$_{72}$
252	GND
253	VDD
254	PIX$_{71}$
255	PIX$_{70}$
256	PIX$_{69}$
257	PIX$_{68}$
258	PIX$_{67}$
259	PIX$_{66}$
260	PIX$_{65}$
261	PIX$_{64}$
262	PIX$_{63}$
263	PIX$_{62}$
264	PIX$_{61}$
265	PIX$_{60}$
266	PIX$_{59}$
267	GND
268	PIX$_{58}$
269	PIX$_{57}$
270	PIX$_{56}$
271	PIX$_{55}$
272	PIX$_{54}$
273	PIX$_{53}$
274	PIX$_{52}$
275	PIX$_{51}$
276	PIX$_{50}$
277	PIX$_{49}$
278	PIX$_{48}$
279	PIX$_{47}$
280	PIX$_{46}$
281	VDD
282	GND
283	PIX$_{45}$
284	PIX$_{44}$
285	PIX$_{43}$
286	PIX$_{42}$
287	PIX$_{41}$
288	PIX$_{40}$
289	PIX$_{39}$
290	PIX$_{38}$
291	PIX$_{37}$
292	PIX$_{36}$
293	PIX$_{35}$
294	PIX$_{34}$
295	PIX$_{33}$
296	PIX$_{32}$
297	PIX$_{31}$
298	PIX$_{30}$
299	PIX$_{29}$
300	VDD
301	PIX$_{28}$
302	PIX$_{27}$
303	PIX$_{26}$
304	GND

Package Drawing - 304 C4FP

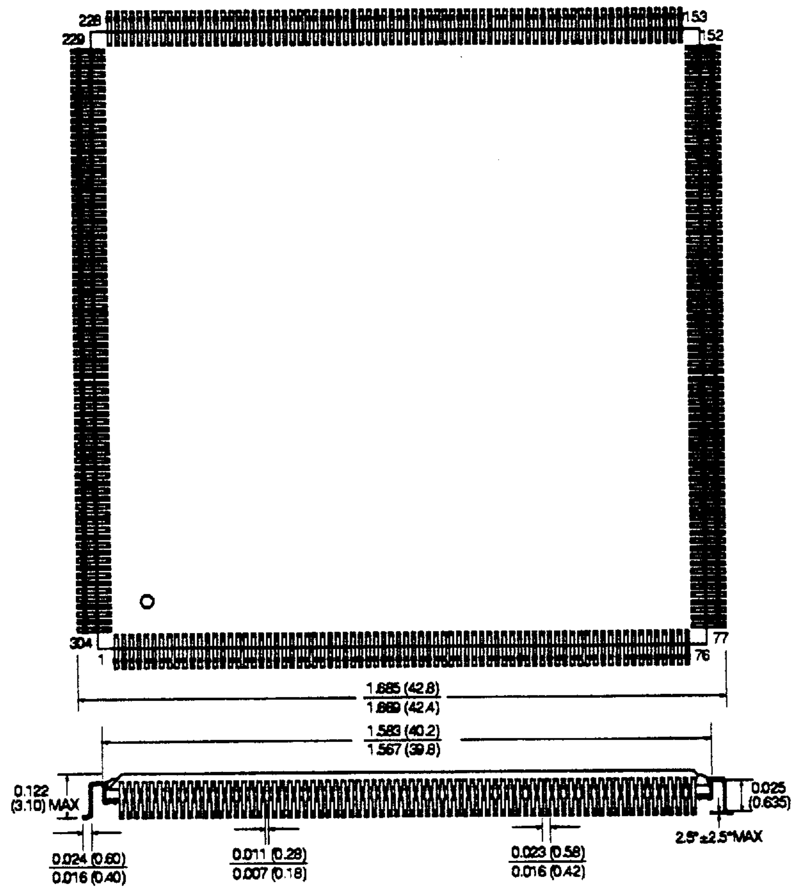


Figure 12. Top View, heat sink not shown. Drawing not to scale in(mm)

ORDERING INFORMATION		
IBM 37RGB561 CF 17	170 MHz	304 C4FP
IBM 37RGB561 CF 22	220 MHz	304 C4FP
IBM 37RGB561 CF 25	250 MHz	304 C4FP

Appendix a. APPLICATION NOTES

Quad Buffering of 8 bit index pixels

With the advanced architecture of the RGB561 it is possible to extend indexed pixel modes to support up to 4 VRAM buffers. Although not obvious at first glance, programming the FB WAT with contradictory parameters can achieve the quad buffering mode. The following table has been modified to help explain the necessary WAT settings.

<table>
<tr><td colspan="11">Table 26. RGB561 Extended Pixel Data Formats</td></tr>
<tr><td colspan="11">For the color modes below, frame buffer data locations are specified in RGB format for frame buffers A/B.</td></tr>
<tr><td>24 bpp</td><td>RED_B 40</td><td>$GREEN_B$ 32</td><td colspan="2">$BLUE_B$ 24</td><td colspan="2">RED_A 16</td><td colspan="2">$GREEN_A$ 8</td><td colspan="2">$BLUE_A$ 0</td></tr>
<tr><td>16 bpp</td><td colspan="2"></td><td>R_B 27</td><td colspan="2">G_B 21</td><td>B_B 16</td><td>R_A 11</td><td colspan="2">G_A 5</td><td>B_A 0</td></tr>
<tr><td>8 bpp</td><td colspan="2"></td><td colspan="2">$INDEX_D$ 24</td><td colspan="2">$INDEX_C$ 16</td><td colspan="2">$INDEX_B$ 8</td><td colspan="2">$INDEX_A$ 0</td></tr>
</table>

In the table above it can be seen that the 8 bit indexed buffers A and B are accessed with conventional WAT settings. Buffer C lines up with the first 8 bits of the 16 bit buffer B pixels. Buffer D lines up with the first 8 bits of the 24 bit buffer B pixels. To access buffers A through D the pixel format bits, buffer select bit and color mode bits (***PIX_FORM, BS and MODE*** in the FB_WAT entries) should be set as follows:

8 bit index buffer	PIX_FORM	BS	MODE
A	00	0	00
B	00	1	00
C	10	1	00
D	11	1	00

It is also recommended that the overlay bits setting (***OVLY***) in Configuration register 1 be set to 11 (Variable WAT control).

The contradictory data is specifying a 16 or 24 bit indexed pixel; the RGB561 only supports 8 bit index mode.

Why it works:

The internal architecture sees the indexed pixel mode and accesses the first 8 bits of data from whatever pixel is presented from the pixel formatting logic. In the case of buffer C the pixel formatter will choose the 16 bits of data assigned to buffer B of the 16 bpp mode, the indexing logic will only use the first 8 bits of this data for accessing the palette. For buffer D the pixel formatter will choose the 24 bits of data assigned to buffer B of the 24 bpp mode, the indexing logic will only use the first 8 bits of this data for accessing the palette.

Multiple frame buffer chroma key

Chroma keying from the overlay frame buffer to the main frame buffer is accommodated in the RGB561 through use of the chroma key registers. By using the double buffer capability of the overlay pixels and the WATs it is possible to select (or chroma key) between three frame buffers. Assume that there are three frame buffers named MAIN, AUX-A and AUX-B that we want to multiplex between based on the pixel value of MAIN. MAIN is an 8 bit frame buffer, AUX-A is a 24 bit frame buffer and AUX-B is a 4 bit frame buffer.

```
IF MAIN = XX
   SELECT = AUX-A
ELSE IF MAIN = YY
   SELECT = AUX-B
ELSE
   SELECT = MAIN
```

To achieve this three way multiplex it is necessary to configure the RGB561 in the 4:1 Extended mode with common 8bit WIDs. The inputs should be wired as follows.

- Wire the MAIN VRAM data bits to the RGB561 Overlay Buffer A pixel inputs (P39 - P32).
- Also wire the MAIN VRAM data bits to the RGB561 WID bits (7 - 0).
- Wire the AUX-A VRAM data bits to the RGB561 Frame Buffer A pixels (P23 - P0).
- Wire the AUX-B VRAM data bits to the RGB561 Overlay Buffer B pixel inputs (P31 - P28).

Load the CHROMA KEY Register #0 to the value XX.

Load the Window Attribute tables with the following data.

ADDRESS	OL_WAT	FB_WAT
location XX	TR = 1	PIX_FORM = 11 BS = 0 MODE = 11 TR = 0
location YY	PIX_FORM = 10 BS = 1 MODE = 00 TR = 0	
All other	PIX_FORM = 00 BS = 0 TR = 0	

Load the Auxillary OVERLAY WAT with the following data.

ADDRESS	AUX_OL_WAT
All	CK/OT = 10 OL = 1

With the MAIN frame buffer wired to the WID inputs the WAT tables will select the 4 bpp AUX-B frame buffer if the MAIN frame buffer pixel is equal to YY. The normal chroma key action will now select the 24 bit AUX-A frame buffer if the value of the MAIN frame buffer pixel is equal to XX. If neither of these values occurs the MAIN frame buffer will be displayed since the OVERLAY WAT is defaulted to buffer A for all pixel values other than XX or YY.

Appendix b. DOCUMENT REVISIONS

REV 1.1

ARCHITECTURE Changes

Display Timing Generator

Removed all information relating to Master/Slave chip operation

REGISTER SUMMARIES Changes

X'0002 Configuration Register 2 (CONF/2)

Renamed bit 3 to CLC (Cursor Location Control)

X'0003 Configuration Register 3 (CONF/3)

Corrected MISR bit definition

X'0020 Display Timing Generator (DTG) Control Register

Renamed to SYNC Control Register. Reserved bits 2, 4 and 6. Removed references to DTG Master Mode.

X'0021 PLL/VCO Divider Register

Included PLL programming equations in standard form. Redefined register bits in terms of programming equations.

X'0022 PLL Reference Register

Redefined register bits in terms of programming equations.

X'0023 AUX_SERIAL_CLOCK Control Register

Deleted.

X'0034-0039 Cursor Location Registers

Split into two groups of registers. First group is the cursor hot spot location registers at ***X'0034-0035***, the second group is the Cursor Location Registers at ***X'0036-0039***.

X'0070-0081 Horizontal/Vertical Registers

The Display Registers have been removed from the document.

X'0082 DTG Timing Reference Register

Renamed to Divided DOT Clock Control Register and bits 0 through 3 are reserved.

SIGNAL PINS Changes

LOAD CLOCK

Removed references to DTG Master and DTG Slave mode.

CSYNC/VSYNC

Removed references to DTG Master and DTG Slave mode. Renamed signal to **CSYNC** Redefined as an input only.

TIMEREF

Removed references to DTG Master and DTG Slave mode. Redefined as an input only. Renamed to **CBLANKIN** for documentation consistency.

CBLANK

Renamed to **CBLANKOUT** for documentation consistency.

SERIAL_CLOCK

Removed references to DTG Master and DTG Slave mode.

AUX_SERIAL_CLOCK

Removed references to being controlled by ***AUX_SERIAL_CLK Control Register*** which has been removed from document.

FIELD

Removed references to DTG Master mode. Redefined as an input only.

AVDD

Renamed as the DAC power supply.

P3_PLL

Renamed as the PLL power supply.

AGND

Renamed as the DAC ground.

P0_PLL

Renamed as the PLL ground.

External Circuitry Changes

Signal Terminations on page 53

Removed pin 31 from table, it is always an input and must be used.

Added pin 198 (FIELD) to table, terminate if unused.

Component Values on page 54

Clarified R8 selection criteria.

Performance Characteristics Changes

AC Characteristics

Removed Figure 9 (SERIAL_CLK and CBLANK output timings).

Corrected Figure 10 (now Figure 8) to show $\overline{\text{CBLANKIN}}$ replacing TIMEREF. Added parameter 25 to diagram. Changed reference to DTG Timing Reference Register to Divided DOT Clock Control Register.

Added parameter 25 to VIDEO Mode Timings, changed parameter 29 to be ***DDOTCLK to*** $\overline{\textbf{CBLANKOUT}}$, changed min and max values for parameters 27 and 29.

DAC Output Levels

Corrected RREF value in Figure 12 (now Figure 10)

PIN List

renamed pin 51 to $\overline{\text{CBLANKOUT}}$ (no function change)

renamed pin 226 to $\overline{\text{CBLANKIN}}$ (no function change)

New Section Added

Appendix a added.

Application Notes section added.

Appendix b added.

Document Revisions section added.