Intel® HD Graphics OpenSource PRM

Volume 3 Part 3: North Display Registers

For the all new 2010 Intel Core Processor Family
Programmer's Reference Manual (PRM)

February 2010

Revision 1.0

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. Intel products are not intended for use in medical, life saving, or life sustaining applications.

Intel may make changes to specifications and product descriptions at any time, without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

The Sandy Bridge chipset family, Havendale/Auburndale chipset family, Intel® 965 Express Chipset Family, Intel® G35 Express Chipset, and Intel® 965GMx Chipset Mobile Family Graphics Controller may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Contents

Revision History

Document Number	Revision Number	Description	Revision Date
IHD-OS-022810-R1V3PR3	1.0	First Release	February 2010

1. North Display Engine Registers

This chapter contains the register descriptions for the display portion of a family of graphics devices. These registers vary by devices within the family of devices, so special attention needs to be paid to which devices use which registers and register fields.

Different devices within the family may add, modify, or delete registers or register fields relative to another device in the same family based on the supported functions of that device.

1.1.1 Terminology

Description	Software Use	Should be implemented as
Read/Write, R/W	This bit can be read or written.	
Reserved	Don't assume a value for these bits. Writes have no effect.	Writes are ignored. Reads return zero.
Reserved: must be zero, MBZ	Software must always write a zero to these bits. This allows new features to be added using these bits that will be disabled when using old software and as the default case.	Writes are ignored. Reads return zero. Maybe be connected as Read/Write in future projects.
Reserved: PBC, software must preserve contents	Software must write the original value back to this bit. This allows new features to be added using these bits.	Read only or test mode Read/Write.
Read Only	This bit is read only. The read value is determined by hardware. Writes to this bit have no effect.	According to each specific bit. The bit value is determined by hardware and not affected by register writes to the actual bit.
Read/Clear, Read/Write Clear	This bit can be read. Writes to it with a one cause the bit to clear.	Hardware events cause the bit to be set and the bit will be cleared on a write operation where the corresponding bit has a one for a value.
Double Buffered	Write when desired. Read gives the unbuffered value (written value) unless specified otherwise. Written values will update to take effect after a certain point. Some have a specific arming sequence where a write to another register is required before the update can take place. This is used to ensure atomic updates of several registers.	Two stages of registers used. First stage is written into and used for readback (unless specified otherwise). First stage value is transferred into second stage at the update point. Second stage value is used to control hardware. Arm/disarm flag for specific arming sequences.

1.1.2 Register Protection for Panel Protection

See the South Display Engine Registers document.

1.1.3 Display Mode Set Sequence

Wait values

FDI training pattern 1 time = 0.5uS
FDI training pattern 2 time = 1.5uS
FDI idle pattern time = 31uS

Enable sequence

1. Enable panel power as needed to retrieve panel configuration
    - a. Enable panel power override using AUX VDD enable override bit
    - b. Wait for delay given in panel requirements
    - c. Leave panel power override enabled until later step
2. Enable PCH clock reference source and PCH SSC modulator, wait for warmup (Can be done anytime before enabling port)
3. If enabling port on PCH: (Must be done before enabling CPU pipe or FDI)
    - a. Enable PCH FDI Receiver PLL, wait for warmup plus DMI latency
    - b. Switch from Rawclk to PCDclk in FDI Receiver (FDI A OR FDI B)
    - c. Enable CPU FDI Transmitter PLL, wait for warmup
4. Enable CPU panel fitter if needed (Can be done anytime before enabling CPU pipe)
5. Configure CPU pipe timings, M/N/TU, and other pipe settings (Can be done anytime before enabling CPU pipe)
6. Enable CPU pipe
7. Configure and enable CPU planes (VGA or hires)
8. If enabling port on PCH:
    - a. Program PCH FDI Receiver TU size same as Transmitter TU size for TU error checking
    - b. Train FDI
        - i. Set pre-emphasis and voltage (iterate if training steps fail)
        - ii. Enable CPU FDI Transmitter and PCH FDI Receiver with training pattern 1 enabled
        - iii. Wait for FDI training pattern 1 time
        - iv. Read PCH FDI Receiver IIR for bit lock in bit 8 (retry at least once if no lock)
        - v. Enable training pattern 2 on CPU FDI Transmitter and PCH FDI Receiver
        - vi. Wait for FDI training pattern 2 time
        - vii. Read PCH FDI Receiver IIR for symbol lock in bit 9 (retry at least once if no lock)
        - viii. Enable normal pixel output on CPU FDI Transmitter and PCH FDI Receiver
        - ix. Wait for FDI idle pattern time for link to become active
    - c. Configure and enable PCH DPLL, wait for PCH DPLL warmup (Can be done anytime before enabling PCH transcoder)
    - d. Configure PCH transcoder timings, M/N/TU, and other transcoder settings (should match CPU settings)
    - e. Enable PCH transcoder
9. Enable panel power through panel power sequencing
10. Wait for panel power sequencing to reach enabled steady state
11. Disable panel power override
12. Enable panel backlight

Disable sequence

1. Disable panel backlight
2. Disable panel power through panel power sequencing
3. Disable CPU planes (VGA or hires)
4. Disable CPU pipe
5. Wait for CPU pipe off status (CPU pipe config register pipe state)
6. Disable CPU panel fitter (Can be done anytime after CPU pipe is off)
7. Else disabling port on PCH:
    a. Disable CPU FDI Transmitter and PCH FDI Receiver
    b. Disable port
    c. Disable PCH transcoder
    d. Wait for PCH transcoder off status (PCH transcoder config register transcoder state)
    e. Disable PCH DPLL (Can be done anytime after PCH ports and transcoder are off)
    f. If no other PCH transcoder is enabled
        i. Switch from PCDclk to Rawclk in PCH FDI Receiver
        ii. Disable CPU FDI Transmitter PLL
        iii. Disable PCH FDI Receiver PLL
8. If SSC is no longer needed, disable PCH SSC modulator
9. If clock reference no longer needed, disable PCH clock reference source

Pipe timings change

Use complete disable sequence followed by complete enable sequence with new mode programmings.
Please note that pipe source size can be changed on the fly when panel fitting is enabled.

Notes

CPU FDI Transmitter should not be set to idle while PCH transcoder is enabled as this will cause PCH transcoder underflow.

1.1.4 Register Instances and Address Offsets

The main body of the register document only contains generic register format information.

The register offset spreadsheet gives the address of each register and which format it is an instance of.

2. North Display Engine Shared Functions

2.1 VGA

The VGA Control register is located here. The remaining VGA registers are located in the VGA Registers document.

2.1.1 VGA_CONTROL—VGA Control

<table>
<tr><th colspan="2">VGA_CONTROL</th></tr>
<tr><td colspan="2">Default Value: 00002900h
Access: R/W
Size (in bits): 32</td></tr>
<tr><td colspan="2">Note: VGA requires panel fitting to be enabled.
Note: VGA is always connected to pipe A.</td></tr>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31</td><td><b>VGA_Display_Disable</b>
Project: All
Default Value: 0b VGA Display Enabled
This bit will disable the VGA compatible display mode. It has no effect on VGA register or A0000-BFFFF memory aperture accesses which are controlled by the PCI configuration and VGA register settings. VGA display should only be enabled if all display planes other than VGA are disabled. After enabling the VGA, most display planes need to stay disabled, only the VGA popup cursor can be enabled.
<b>Note</b>: The VGA SR01 screen off bit must be programmed when enabling and disabling VGA. See the VGA Registers document.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>Enable</td><td>VGA Display Enabled</td><td>All</td></tr>
<tr><td>1b</td><td>Disable</td><td>VGA Display Disabled</td><td>All</td></tr>
</table>
</td></tr>
<tr><td>30:27</td><td><b>Reserved</b> Project: All Format: PBC</td></tr>
</table>

VGA_CONTROL

26 **VGA_Border_Enable**

Project: All

Default Value: 0b

This bit determines if the VGA border areas are included in the active display area. The border will be scaled along with the pixel data. The VGA popup cursor can be positioned overlapping the border area of the image.

Value	Name	Description	Project
0b	Disable	VGA border areas are not displayed	All
1b	Enable	VGA border areas are displayed	All

25 **Reserved** Project: All Format: PBC

24 **Pipe_CSC_Enable**

Project: All

Default Value: 0b

This bit enables pipe color space conversion for the VGA pixel data. CSC mode in the pipe CSC registers must be set to match the format of the VGA pixel data.

Value	Name	Description	Project
0b	Bypass	VGA pixel data bypasses the pipe color space conversion logic	All
1b	Pass	VGA pixel data passes through the pipe color space conversion logic	All

23:21 **Reserved** Project: All Format: PBC

20 **Legacy_8Bit_Palette_En**

Project: All

Default Value: 0b

This bit only affects reads and writes to the palette through VGA I/O addresses. In the 6-bit mode, the 8-bits of data are shifted up two bits on the write (upper two bits are lost) and shifted two bits down on the read. It provides backward compatibility for original VGA programs (in it's default state) as well as VESA VBE support for 8-bit palette. It does not affect palette accesses through the palette register MMIO path.

Value	Name	Description	Project
0b	6 bit DAC	6-bit DAC	All
1b	8 bit DAC	8-bit DAC	All

19 **Palette_Bypass**

Project: All

Security: Test

Default Value: 0b

Value	Name	Description	Project
0b	Pass	Pass VGA data through the palette	All
1b	Bypass	Bypass the palette	All

VGA_CONTROL

18 **Nine_Dot_Disable**

Project: All

Security: Test

Default Value: 0b

Prevents DOS applications from setting the VGA display into a real 9-dot per character operation mode, instead the device emulates that using 8-dots per character. The VGA register bit SR01<0> functionality is disabled. VGA panning control handles the pseudo 9-dot mode when both this bit is set and SR01<0> is clear.

Value	Name	Description	Project
0b	9 dot	Enable use of 9-dot enable bit in VGA registers	All
1b	8 dot	Ignore the 9-dot per character bit and always use 8	All

17:16 **Reserved** Project: All Format: PBC

15:12 **VGA_DE_throttling**

Project: All

Security: Test

Default Value: 0010b 33%

These bits throttle the VGA engine's display pipe line from generating pixels too quickly during the display enable region. Throttling should be set at boot time.

Value	Name	Description	Project
0000b,1000b	0%	0% (Disable)	All
0001b,1001b	50%	50%	All
0010b	33%	33% (Default)	All
0011b	25%	25%	All
0100b	20%	20%	All
0101b	17%	17%	All
0110b	15%	15%	All
0111b	10%	10%	All
1010b	66%	66%	All
1011b	75%	75%	All
1100b	80%	80%	All
1101b	90%	90%	All
1110b	85%	85%	All
1111b	82%	82%	All

VGA_CONTROL

11:8

VGA_blank_throttling_blank

Project: All

Security: Test

Default Value: 1001b 50%

These bits throttle the VGA engine's display pipe line from generating pixels too quickly during the blanking region. Throttling should be set at boot time.

Value	Name	Description	Project
0000b,1000b	0%	0% (Disable)	All
0001b,1001b	50%	50% (default)	All
0010b	33%	33%	All
0011b	25%	25%	All
0100b	20%	20%	All
0101b	17%	17%	All
0110b	15%	15%	All
0111b	10%	10%	All
1010b	66%	66%	All
1011b	75%	75%	All
1100b	80%	80%	All
1101b	90%	90%	All
1110b	85%	85%	All
1111b	82%	82%	All

7:6

Blink_Duty_Cycle

Project: All

Default Value: 00b

Controls the VGA text mode blink duty cycle relative to the VGA cursor blink duty cycle.

Value	Name	Description	Project
00b	100%	100% Duty Cycle, Full Cursor Rate	All
01b	25%	25% Duty Cycle, ½ Cursor Rate	All
10b	50%	50% Duty Cycle, ½ Cursor Rate	All
11b	75%	75% Duty Cycle, ½ Cursor Rate	All

5:0

VSYNC_Blink_Rate Project: All

Controls the VGA blink rate in terms of the number of VSYNCs per on/off cycle. These bits are programmed with the (VSYNCs/cycle)/2-1. The proper programming of this register is determined by the VSYNC rate that the display requires when in a VGA display mode.

2.2 Frame Buffer Compression

2.2.1 FBC_CFB_BASE—FBC Compressed Buffer Address

FBC_CFB_BASE

Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W**
Size (in bits):	**32**

The contents of this register can not be changed while compression is enabled.

Bit	Description
31:28	**Reserved** Project: All Format: MBZ
27:12	**CFB_Offset_Address** Project: All This register specifies offset of the Compressed Frame Buffer from the base of stolen memory. The buffer must be 4K byte aligned.
11:0	**Reserved** Project: All Format: MBZ

2.2.2 FBC_CTL— FBC Control

FBC_CTL

Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W**
Size (in bits):	**32**

The contents of this register can not be changed, except bit 31, while compression is enabled.
Frame Buffer Compression is only supported with 16bpp and 32bpp 8:8:8 primary plane source pixel formats. It is not supported with any 10:10:10 or 64bpp format.

<table>
<tr><th colspan="2">FBC_CTL</th></tr>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31</td><td><b>Enable_FBC</b>
Project: All
Default Value: 0b
This bit is used to globally enable FBC function at the next Vertical Blank start.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>Disable</td><td>Disable frame buffer compression</td><td>All</td></tr>
<tr><td>1b</td><td>Enable</td><td>Enable frame buffer compression</td><td>All</td></tr>
</table></td></tr>
<tr><td>30:29</td><td><b>Plane_Select</b>
Project: All
Default Value: 00b
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>00b</td><td>Primary Plane A</td><td>Primary Plane A</td><td>All</td></tr>
<tr><td>01b</td><td>Primary Plane B</td><td>Primary Plane B</td><td>All</td></tr>
<tr><td>10b</td><td>Primary Plane C</td><td>Primary Plane C</td><td>All</td></tr>
<tr><td>11b</td><td>Reserved</td><td>Reserved</td><td>All</td></tr>
</table></td></tr>
<tr><td>28</td><td><b>CPU_Fence_Enable</b>
Project: All
Default Value: 0b
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>No CPU Disp Buf</td><td>Display Buffer is not in a CPU fence. No modifications are allowed from CPU to the Display Buffer</td><td>All</td></tr>
<tr><td>1b</td><td>CPU Disp Buf</td><td>Display Buffer exists in a CPU fence</td><td>All</td></tr>
</table></td></tr>
<tr><td>27:25</td><td><b>Reserved</b> Project: All Format: MBZ</td></tr>
<tr><td></td><td></td></tr>
<tr><td>15</td><td><b>SLB_Init_Flush_Disable</b>
Project: All
Security: Test
Default Value: 0b
Setting this bit will disable the SLB flush mechanism for the first frame FBC is on.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>Enable</td><td>Enable the SLB initialization flush</td><td>All</td></tr>
<tr><td>1b</td><td>Disable</td><td>Disable SLB initialization flush</td><td>All</td></tr>
</table></td></tr>
<tr><td>14:11</td><td><b>Reserved</b> Project: All Format: MBZ</td></tr>
</table>

FBC_CTL

7:6 **Compression_Limit**

Project: All

Default Value: 0b

This register sets a minimum limit on compression. It is also used to determine the size of the compressed buffer.

Compression Ratio	Pixel Format	
	16 bpp	32 bpp
1	Not Supported	Supported (CFB=FB)
½	Supported (CFB=FB)	Supported (CFB=1/2 FB)
¼	Supported (CFB=1/2FB)	Supported (CFB=1/4 FB)

FB = Frame Buffer Size

CFB = Compressed Frame Buffer Size

Value	Name	Description	Project
00b	1:1	1:1 compression, compressed buffer is the same size as the uncompressed buffer	All
01b	2:1	2:1 compression, compressed buffer is one half the size of the uncompressed buffer.	All
10b	4:1	4:1 compression, compressed buffer is one quarter the size of the uncompressed buffer.	All
11b	Reserved	Reserved	All

5:4 **Write_Back_Watermark**

Project: All

Default Value: 0b

The compressed data write back engine waits for this number of entries to be ready before writing the data out to memory.

Value	Name	Description	Project
00b	4	4 entries	All
01b	8	8 entries	All
10b	16	16 entries	All
11b	32	32 entries	All

3:0 **CPU_Fence_Number**

Project: All

Default Value: 0b

This field specifies the CPU visible FENCE number corresponding to the placement of the uncompressed frame buffer.

[DevIVB] iMPH will only send the host modify message when modifications are in the fence selected in the DPFC_CONTROL_SA register CPUFNCNUM field. The fence field in the FBC Host Modification message will always be 0 and this field must be programmed to 0 to match.

2.2.3 FBC_RECOMP_CTL — FBC ReComp Control

<table>
<tr><th colspan="2">FBC_RECOMP_CTL</th></tr>
<tr><td colspan="2"><b>Register Type:</b> MMIO
<b>Project:</b> All
<b>Default Value:</b> 00000000h
<b>Access:</b> R/W
<b>Size (in bits):</b> 32</td></tr>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31:28</td><td><b>Reserved</b> Project: All Format: MBZ</td></tr>
<tr><td>27</td><td><b>Enable_ReComp_Stall</b>
Project: All
Default Value: 0b
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>Disable</td><td>Disable</td><td>All</td></tr>
<tr><td>1b</td><td>Enable</td><td>Enable</td><td>All</td></tr>
</table></td></tr>
<tr><td>26:16</td><td><b>ReComp_Invalidation_Watermark</b> Project: All
If this many or more invalidations occur in one frame, stop compression until the number falls below watermark, then start the recomp timer.</td></tr>
<tr><td>15:6</td><td><b>Reserved</b> Project: All Format: MBZ</td></tr>
<tr><td>5:0</td><td><b>ReComp_Timer_Count</b> Project: All
After invalidations fall below watermark, wait this many frames before restarting the compressor.
A 0 means restart compression on the following frame.</td></tr>
</table>

2.3 Interrupts

2.3.1 Display Engine Interrupt Bit Definition

Display Engine Interrupt Bit Definition

Project:	**All**
Size (in bits):	**32**

Display Engine (DE) interrupt bits come from events within the display engine, except for some which explicitly list a non-display engine source. The DE_IIR and GT_IIR and PM_IIR are ORed together to generate the CPU interrupt.

The Display Engine Interrupt Control Registers all share the same bit definitions from this table.

Bit	Description
31	**Master_Interrupt_Control** Project: All Format: This bit exists only in the DEIER Display Engine Interrupt Enable Register. This is the master control for the Display to CPU interrupt. This bit must be set to 1 for any interrupts to propagate to the system.
29	**GSE** Project: All Format: This is an active high pulse on the GSE system level event.
28	**PCH_Display_interrupt_event** Project: All Format: This is an active high level while there is an interrupt being generated by the PCH Display. It will stay asserted until the interrupts in the PCH Display are all cleared.
26	**AUX_Channel_A** Project: All Format: This is an active high pulse on the AUX A done event.
25	**DPST_histogram_event** Project: All Format: This is an active high pulse on the DPST histogram event.
24	**DPST_phase_in_event** Project: All Format: This is an active high pulse on the DPST phase in event.
23:15	**Reserved** Project: All Format:
14	**Sprite_Plane_Flip_Done_C** Project: All Format: This is an active high pulse when a sprite plane flip is done.
13	**Primary_Plane_Flip_Done_C** Project: All Format: This is an active high pulse when a primary plane flip is done.
12	**Line_Compare_Pipe_C** Project: All Format: This is an active high level for the duration of the selected pipe scan line.
11	**Vsync_Pipe_C** Project: All Format: This is an active high level for the duration of the pipe vertical sync.
10	**Vblank_Pipe_C** Project: All Format: This is an active high level for the duration of the pipe vertical blank.
9	**Sprite_Plane_Flip_Done_B** Project: All Format: This is an active high pulse when a sprite plane flip is done.

Display Engine Interrupt Bit Definition

8	**Primary_Plane_Flip_Done_B** This is an active high pulse when a primary plane flip is done.	Project:	All	Format:
7	**Line_Compare_Pipe_B** This is an active high level for the duration of the selected pipe scan line.	Project:	All	Format:
6	**Vsync_Pipe_B** This is an active high level for the duration of the pipe vertical sync.	Project:	All	Format:
5	**Vblank_Pipe_B** This is an active high level for the duration of the pipe vertical blank.	Project:	All	Format:
4	**Sprite_Plane_Flip_Done_A** This is an active high pulse when a sprite plane flip is done.	Project:	All	Format:
3	**Primary_Plane_Flip_Done_A** This is an active high pulse when a primary plane flip is done.	Project:	All	Format:
2	**Line_Compare_Pipe_A** This is an active high level for the duration of the selected pipe scan line.	Project:	All	Format:
1	**Vsync_Pipe_A** This is an active high level for the duration of the pipe vertical sync.	Project:	All	Format:
0	**Vblank_Pipe_A** This is an active high level for the duration of the pipe vertical blank.	Project:	All	Format:

2.3.2 GT Interrupt Bit Definition

GT Interrupt Bit Definition

Project: **All**
Size (in bits): **32**

GT interrupt bits come to display through the GT interrupt message. The DE_IIR and GT_IIR and PM_IIR are ORed together to generate the CPU interrupt.
The GT Interrupt Control Registers all share the same bit definitions from this table.

Bit	Description
31	**Reserved** Project: All Format:
30	**Blitter_AS_Context_Switch_Interrupt** Project: All This is a write of logic1 via GT interrupt message bit 30
29	**Blitter_page_directory_faults** Project: All This is a write of logic1 via GT interrupt message bit 29

GT Interrupt Bit Definition	
28:27	**Reserved** Project: All Format:
26	**Blitter_MI_FLUSH_DW_notify** Project: All This is a write of logic1 via GT interrupt message bit 26
25	**Blitter_Command_Streamer_error_interrupt** Project: All This is a write of logic1 via GT interrupt message bit 25
24	**Billter_MMIO_sync_flush_status** Project: All This is a write of logic1 via GT interrupt message bit 24
23	**Reserved** Project: All Format:
22	**Blitter_Command_Streamer_MI_USER_INTERRUPT** Project: All This is a write of logic1 via GT interrupt message bit 22
21	**Reserved** Project: All Format:
19	**Video_page_directory_faults** Project: All This is a write of logic1 via GT interrupt message bit 19
18	**Video_Command_Streamer_Watchdog_counter_exceeded** Project: All This is a write of logic1 via GT interrupt message bit 18
17	**Reserved** Project: All Format:
16	**Video_MI_FLUSH_DW_notify** Project: All This is a write of logic1 via GT interrupt message bit 16
15	**Video_Command_Streamer_error_interrupt** Project: All This is a write of logic1 via GT interrupt message bit 15
14	**Video_MMIO_sync_flush_status** Project: All This is a write of logic1 via GT interrupt message bit 14
13	**Reserved** Project: All Format:
12	**Video_Command_Streamer_MI_USER_INTERRUPT** Project: All This is a write of logic1 via GT interrupt message bit 12
11:9	**Reserved** Project: All Format:

GT Interrupt Bit Definition	
8	**Render_AS_Context_Switch_Interrupt** Project: All This is a write of logic1 via GT interrupt message bit 8
7	**Render_page_directory_faults** Project: All This is a write of logic1 via GT interrupt message bit 7
6	**Render_Command_Streamer_Watchdog_counter_exceeded** Project: All This is a write of logic1 via GT interrupt message bit 6
5	**Reserved** Project: All Format:
4	**Render_PIPE_CONTROL_notify** Project: All This is a write of logic1 via GT interrupt message bit 4
3	**Render_Command_Streamer_error_interrupt** Project: All This is a write of logic1 via GT interrupt message bit 3
2	**Render_MMIO_sync_flush_status** Project: All This is a write of logic1 via GT interrupt message bit 2
0	**Render_Command_Streamer_MI_USER_INTERRUPT** Project: All This is a write of logic1 via GT interrupt message bit 0

2.3.3 Power Management Interrupt Bit Definition

Power Management Interrupt Bit Definition	
Project: **All** **Size(in bits):** **32**	
Power Management interrupt bits come to display through the PM interrupt message. The DE_IIR and GT_IIR and PM_IIR are ORed together to generate the CPU interrupt. The Power Management Interrupt Control Registers all share the same bit definitions from this table.	
Bit	**Description**
31:26	**Reserved** Project: All Format: MBZ

Power Management Interrupt Bit Definition

25	**PCU_pcode2driver_mailbox_event** Project: All Format: This is a write of logic1 via PM interrupt message bit 25
24	**PCU_Thermal_Event** Project: All Format: This is a write of logic1 via PM interrupt message bit 24
23:7	**Reserved** Project: All Format: MBZ
6	**Render_Frequency_Downward_Timeout_During_RC6_interrupt** Project: All Format: This is a write of logic1 via PM interrupt message bit 6
5	**RP_UP_threshold_interrupt** Project: All Format: This is a write of logic1 via PM interrupt message bit 5
4	**RP_DOWN_threshold_interrupt** Project: All Format: This is a write of logic1 via PM interrupt message bit 4
3	**Render_geyserville_Controller_disable_state_interrupt** Project: All Format: This is a write of logic1 via PM interrupt message bit 3
2	**Render_geyserville_UP_evaluation_interval_interrupt** Project: All Format: This is a write of logic1 via PM interrupt message bit 2
1	**Render_geyserville_Down_evaluation_interval_interrupt** Project: All Format: This is a write of logic1 via PM interrupt message bit 1
0	**Reserved** Project: All Format: MBZ

2.3.4 ISR — Interrupt Status

ISR	
Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**Read Only**
Size (in bits):	**32**

See the interrupt bit definition tables to find the source event for each interrupt bit.

Bit	Description
31:0	**Interrupt_Status_Bits** Project: All This field contains the non-persistent values of all interrupt status bits. The IMR register selects which of these interrupt conditions are reported in the persistent IIR.

Value	Name	Description	Project
0b	Condition Doesn't exist	Interrupt Condition currently does not exist	All
1b	Condition Exists	Interrupt Condition currently exists	All

Programming Notes
Some inputs to this register are short pulses; therefore software should not expect to use this register to sample these conditions.

2.3.5 IMR — Interrupt Mask

<table>
<tr><th colspan="2">IMR</th></tr>
<tr><td colspan="2">Register Type: MMIO
Project: All
Default Value: FFFFFFFFh
Access: R/W
Size (in bits): 32</td></tr>
<tr><td colspan="2">For GT command streamer interrupts DO NOT use this register to mask interrupt events. Instead use the individual GT command streamer MASK bits.
For PM interrupts DO NOT use this register to mask interrupt events. Instead use the individual PM MASK bits in the corresponding PMunit register space.
<b>See the interrupt bit definition tables to find the source event for each interrupt bit.</b></td></tr>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31:0</td><td><b>Interrupt_Mask_Bits</b>
Project: All
This field contains a bit mask which selects which interrupt bits from the ISR are reported in the IIR.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>Not Masked</td><td>Not Masked – will be reported in the IIR</td><td>All</td></tr>
<tr><td>1b</td><td>Masked</td><td>Masked – will not be reported in the IIR</td><td>All</td></tr>
</table>
</td></tr>
</table>

2.3.6 IIR — Interrupt Identity

IIR

Register Type: MMIO
Project: All
Default Value: 00000000h
Access: R/W Clear
Size (in bits): 32

See the interrupt bit definition tables to find the source event for each interrupt bit.

Bit	Description
31:0	**Interrupt_Identity_Bits** Project: All This field holds the persistent values of the interrupt bits from the ISR which are unmasked by the IMR. If enabled by the IER, bits set in this register will generate a CPU interrupt. Bits set in this register will remain set (persist) until the interrupt condition is cleared by writing a '1' to the appropriate bits. For each bit, the IIR can store a second pending interrupt if two or more of the same interrupt conditions occur before the first condition is cleared. Upon clearing the interrupt, the IIR bit will momentarily go low, then return high to indicate there is another interrupt pending

Value	Name	Description	Project
0b	Condition Not Detected	Interrupt Condition Not Detected	All
1b	Condition Detected	Interrupt Condition Detected (may or may not have generated a CPU interrupt)	All

2.3.7 IER — Interrupt Enable

IER

Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W**
Size (in bits):	**32**

See the interrupt bit definition tables to find the source event for each interrupt bit.

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31:0</td><td><b>Interrupt_Enable_Bits</b>
Project: All
The bits in this register enable a CPU interrupt to be generated whenever the corresponding bit in the IIR becomes set. A disabled interrupt will still appear in the IIR register to allow polling of interrupt sources. The DE_IER master interrupt control bit must be set to 1 for any interrupts to propagate to the system.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>Disable</td><td>Disable</td><td>All</td></tr>
<tr><td>1b</td><td>Enable</td><td>Enable</td><td>All</td></tr>
</table>
</td></tr>
</table>

2.3.8 HOTPLUG_CTL — Hot Plug Control

HOTPLUG_CTL	
Register Type: Project: Default Value: Access: Size (in bits):	MMIO All 00000000h R/W 32
Bit	**Description**
31:5	**Reserved** Project: All Format:

2.4 Display Engine Render Response

2.4.1 Display Engine Render Response Message Bit Definition

Display Engine Render Response Message Bit Definition

Project: **All**
Size(in bits): **32**

Display Engine (DE) render response message bits come from events within the display engine. The Display Engine Render Response Message Registers all share the same bit definitions from this table.

Bit	Description
31:23	**Reserved** Project: All Format: MBZ
22	**Pipe_C_Start_of_Horizontal_Blank_Event** Project: All Format: This even will be reported on the start of the Pipe C Horizontal Blank.
21	**Pipe_C_Start_of_Vertical_Blank_Event** Project: All Format: This even will be reported on the start of the Pipe C Vertical Blank.
20	**Pipe_C_Sprite_Plane_Flip_Done_Event** Project: All Format: This even will be reported on the completion of a flip for the Pipe C Sprite Plane.
19:16	**Reserved** Project: All Format: MBZ
15	**Pipe_C_Primary_Plane_Flip_Done_Event** Project: All Format: This even will be reported on the completion of a flip for the Pipe C Primary Plane.
14	**Pipe_C_Scanline_Event** Project: All Format: This even will be reported on the start of the scan line specified in the Pipe C Display Scan Line Count Range Compare Register.
13	**Pipe_B_Start_of_Horizontal_Blank_Event** Project: All Format: This even will be reported on the start of the Pipe B Horizontal Blank.
12	**Reserved** Project: All Format: MBZ
11	**Pipe_B_Start_of_Vertical_Blank_Event** Project: All Format: This even will be reported on the start of the Pipe B Vertical Blank.
10	**Pipe_B_Sprite_Plane_Flip_Done_Event** Project: All Format: This even will be reported on the completion of a flip for the Pipe B Sprite Plane.
9	**Pipe_B_Primary_Plane_Flip_Done_Event** Project: All Format: This even will be reported on the completion of a flip for the Pipe B Primary Plane.
8	**Pipe_B_Scanline_Event** Project: All Format: This even will be reported on the start of the scan line specified in the Pipe B Display Scan Line Count Range Compare Register.
7:6	**Reserved** Project: All Format: MBZ

Display Engine Render Response Message Bit Definition	
5	**Pipe_A_Start_of_Horizontal_Blank_Event** Project: All Format: This even will be reported on the start of the Pipe A Horizontal Blank.
4	**Reserved** Project: All Format: MBZ
3	**Pipe_A_Start_of_Vertical_Blank_Event** Project: All Format: This even will be reported on the start of the Pipe A Vertical Blank.
2	**Pipe_A_Sprite_Plane_Flip_Done_Event** Project: All Format: This even will be reported on the completion of a flip for the Pipe A Sprite Plane.
1	**Pipe_A_Primary_Plane_Flip_Done_Event** Project: All Format: This even will be reported on the completion of a flip for the Pipe A Primary Plane.
0	**Pipe_A_Scanline_Event** Project: All Format: This even will be reported on the start of the scan line specified in the Pipe A Display Scan Line Count Range Compare Register.

2.4.2 DE_RRMR — Display Engine Render Response Mask

<table>
<tr><th colspan="2">DE_RRMR</th></tr>
<tr><td colspan="2">Register Type: MMIO
Project: All
Default Value: FFFFFFFFh
Access: R/W
Size (in bits): 32</td></tr>
<tr><td colspan="2"><b>See the render response message bit definition table to find the source event for each bit.</b></td></tr>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31:0</td><td><b>DE_RRMR</b>
Project: All
Format: Display Engine Render Response Message Bit Definition See Description Above
This field contains a bit mask which selects which events cause and are reported in the render response message.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>Not Masked</td><td>Not Masked – will cause and be reported in the message</td><td>All</td></tr>
<tr><td>1b</td><td>Masked</td><td>Masked – will not cause or be reported in the message</td><td>All</td></tr>
</table></td></tr>
</table>

2.5 Display Timestamp

This register provides an elapsed real-time value that can be used as a timestamp for GPU events over short periods of time.

The register counts at a constant frequency by adjusting the increment amount according to the actual core display clock frequency. SW therefore does not need to know the reference clock frequency.

2.5.1 TIMESTAMP_CTR—Time Stamp Counter Value

TIMESTAMP_CTR	
Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W Clear**
Size (in bits):	**32**
Bit	**Description**
31:0	**TIMESTAMP_Counter** Project: All Format: This field increments every microsecond. The value in this field is latched in the Pipe Flip TIMESTAMP registers when flips occur, and in the Pipe Frame TIMESTAMP registers at start of vertical blank. The register value will reset if any value is written to it. The register is **not** reset by a graphics software reset.

2.5.2 TIMESTAMP_LOAD—Time Stamp Load Value

TIMESTAMP_LOAD	
Register Type:	**MMIO**
Project:	**All**
Security:	**Test**
Default Value:	**00000000h**
Access:	**R/W**
Size (in bits):	**32**
Double Buffer Update Point:	**PSMI wipe**
Bit	**Description**
31:0	**TIMESTAMP_Load** Project: All Format: The value written to this register will load into the timestamp counter at the next PSMI wipe. The value read from this register is the timestamp counter value from the previous PSMI wipe.

2.6 Display Arbitration Control

2.6.1 ARB_CTL—Display Arbitration Control 1

ARB_CTL	
Register Type: MMIO Project: All Default Value: D6661056h Access: R/W Size (in bits): 32	
Bit	**Description**
31	**FBC_Memory_Wake** Project: All Security: Test Default Value: 1b On Setting this bit allows FBC compressed write requests to wake memory from SR.
30	**KVMr_Memory_Wake** Project: All Security: Test Default Value: 1b On Setting this bit allows KVMr display write back requests to wake memory from SR.
29	**Opportunistic_Fetch_Enable** Project: All Security: Test Default Value: 0b Setting this bit allows opportunistic data fetches (even when above watermark) when other clients wake the system from SelfRefresh. For any opportunistic fetch to happen, display should not be in the process of waking the system.
28:26	**HP_Queue_Watermark** Project: All Default Value: 101b 6 The value in this register indicates the number of entries the high priority queue should have before it can be read. The value is zero based.

<table>
<tr><th colspan="2">ARB_CTL</th></tr>
<tr><td>25:24</td><td><b>LP_Write_Request_Limit</b>
Project: All
Default Value: 10b 4
The value in this register indicates the maximum number of back to back LP write requests that will be accepted from a single client before re-arbitrating.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>00b</td><td>1</td><td>1</td><td>All</td></tr>
<tr><td>01b</td><td>2</td><td>2</td><td>All</td></tr>
<tr><td>10b</td><td>4</td><td>4 (default)</td><td>All</td></tr>
<tr><td>11b</td><td>8</td><td>8</td><td>All</td></tr>
</table></td></tr>
<tr><td>23:20</td><td><b>TLB_Request_Limit</b>
Project: All
Default Value: 0110b 6
Range: 1..15
The value in this register indicates the maximum number of TLB requests that can be made in an arbitration loop. Zero is not a valid programming.</td></tr>
<tr><td>19:16</td><td><b>TLB_Request_InFlight_Limit</b>
Project: All
Default Value: 0110b 6
Range: 1..15
The value in this register indicates the maximum number of TLB (or VTd) requests that can be in flight at any given time. Zero is not a valid programming.</td></tr>
<tr><td>15</td><td><b>FBC_Watermark_Disable</b>
Project: All
Security: Test
Default Value: 0b
Setting this bit disables the FBC watermarks.</td></tr>
<tr><td>14:13</td><td><b>Tiled_Address_Swizzling</b>
Project: All
Default Value: 00b
DRAM configuration registers show if memory address swizzling is needed.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>00b</td><td>No Display</td><td>No display request address swizzling</td><td>All</td></tr>
<tr><td>01b</td><td>Enable</td><td>Enable display request address bit[6] swizzling for tiled surfaces</td><td>All</td></tr>
<tr><td>1Xb</td><td>Reserved</td><td>Reserved</td><td>All</td></tr>
</table></td></tr>
</table>

ARB_CTL	
12:8	**HP_Page_Break_Limit** Project: All Default Value: 10000b 16 Range: 1..31 The value in this register represents the maximum number of page breaks allowed in a HP request chain. Zero is not a valid programming.
7	**Reserved** Project: All Format:
6:0	**HP_Data_Request_Limit** Project: All Default Value: 1010110b 86 Range: 1..127 The value in this register represents the maximum number of cachelines allowed in a HP request chain. Zero is not a valid programming.

2.6.2 ARB_CTL2—Display Arbitration Control 2

<table>
<tr><th colspan="2">ARB_CTL2</th></tr>
<tr><td colspan="2"><b>Register Type:</b> MMIO
<b>Project:</b> All
<b>Default Value:</b> 00000000h
<b>Access:</b> R/W
<b>Size (in bits):</b> 32</td></tr>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31:9</td><td><b>Reserved</b> Project: All Format:</td></tr>
<tr><td>8</td><td><b>Fetch_Timing</b>
Project: All
Default Value: 0b
The value in this register is valid only when Opportunistic Fetches are enabled. The value in this register is used to specify when an opportunistic fetch can happen. For any opportunistic fetch to happen, display should not be in the process of waking the system.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>FE inSR</td><td>Fetch on falling edge of inSR</td><td>All</td></tr>
<tr><td>1b</td><td>Not inSR</td><td>Fetch when not inSR</td><td>All</td></tr>
</table>
</td></tr>
</table>

ARB_CTL2

7 **Opportunistic_Fetch_Behavior**

Project: All

Default Value: 0h

The value in this register is valid only when Opportunistic Fetches are enabled. The value in this register represents the fetch behavior when an opportunistic fetch is triggered. For any opportunistic fetch to happen, display should not be in the process of waking the system.

Value	Name	Description	Project
0h	One Burst	One Burst Only	All
1h	Fill FIFO	Fill FIFO to Top	All

6 **Data_Buffer_Partitioning**

Project: All

Default Value: 0b

This bit controls the data buffer partitioning when sprite LP states are used.

Value	Name	Description	Project
0b	1/2	Sprite has 1/2 and primary has 1/2 of the buffer	All
1b	5/6	Sprite has 5/6 and primary has 1/6 of the buffer	All

5:4 **Inflight_HP_Read_Request_Limit**

Project: All

Default Value: 00b

The value in this register represents the maximum number of HP read request transactions that can inflight at any given time.

Value	Name	Description	Project
00b	128 HP	128 HP inflight transactions limit	All
01b	64 HP	64 HP inflight transactions limit	All
10b	32 HP	32 HP inflight transactions limit	All
11b	16 HP	16 HP inflight transactions limit	All

3:2 **Reserved** Project: All Format:

1:0 **RTID_FIFO_Watermark**

Project: All

Default Value: 0b

The value in this register represents the watermark value for the RTID FIFO. HP transactions will start only when the FIFO level is above or equal the watermark

Value	Name	Description	Project
00b	8 RTIDs	8 RTIDs available in FIFO	All
01b	16 RTIDs	16 RTIDs available in FIFO	All
10b	32 RTIDs	32 RTIDs available in FIFO	All
11b	Reserved	Reserved	All

2.7 Display Watermarks

The watermark registers are used to control the display to memory request timing. The watermarks must be programmed according to the rules provided in the “Programming Watermarks” document. The default values of the watermarks should allow the display to operate in any high power mode supported by the memory configuration. However, the default watermarks are not optimized for power or memory bandwidth efficiency.

Watermarks must enable from the bottom up, meaning if WM_LP2 is disabled, WM_LP3 must also be disabled, and if WM_LP1 is disabled, both WM_LP2 and WM_LP3 must also be disabled. Watermark latency values must increase from the bottom up, meaning WM_LP1 (if enabled) must have higher latency than WM_PIPE, and so on.

2.7.1 WM_PIPE—Pipe Main Watermarks

WM_PIPE	
Register Type:	**MMIO**
Project:	**All**
Default Value:	**00783818h**
Access:	**R/W**
Size (in bits):	**32**

These are the normal watermark values which are used when display is not in any Low Power (LP) state.

Bit	Description
31:23	**Reserved** Project: All Format:
22:16	**Pipe_Primary_Watermark** Project: All Number in 64Bs of data in FIFO below which the Pipe Primary Plane stream will generate requests to memory
15	**Reserved** Project: All Format:
14:8	**Pipe_Sprite_Watermark** Project: All Number in 64Bs of data in FIFO below which the Pipe Sprite Plane stream will generate requests to memory
7:6	**Reserved** Project: All Format:
5:0	**Pipe_Cursor_Watermark** Project: All Number in 64Bs of data in FIFO below which the Pipe Cursor Plane stream will generate requests to memory

2.7.2 WM_LP—Low Power Watermarks

The Low Power (LP) watermark register will be used when only one pipe is enabled, sprite scaling is not enabled, and the power controller has requested display go into the LP state.

WM_LP

Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W**
Size (in bits):	**32**

These are Low Power (LP) watermark values which will be used when display is in a LP state.

Bit	Description
31	**Enabled** Project: All Enables this LP watermark. This bit allows the associated LP state to be used.
30:24	**Latency** Project: All The latency associated with this LP watermark in half usecs.
23:20	**FBC_LP_Watermark** Project: All Number of equivalent lines of the primary display for this watermark
19:18	**Reserved** Project: All Format:
17:8	**LP_Primary_Watermark** Project: All Number in 64Bs of data in the display data buffer below which the Primary Plane stream will generate requests to memory.
7:0	**LP_Cursor_Watermark** Project: All Number in 64Bs of data in the display data buffer below which the Cursor Plane stream will generate requests to memory.

2.7.3 WM_LP_SPR—Low Power Sprite Watermark

The Low Power Sprite (LP_SPR) watermark register will be used when one pipe is enabled, a sprite is enabled, sprite scaling is not enabled, and the power controller has requested display go into the LP state. This will be used together with the associated LP watermarks for FBC, Primary, and Cursor.

WM_LP_SPR

Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W**
Size (in bits):	**32**

This is a Low Power Sprite (LP_SPR) watermark value which will be used when display is in a LP state.

Bit	Description
31:10	**Reserved** Project: All Format:
9:0	**LP_Sprite_Watermark** Project: All Default Value: 0b Number in 64Bs of data in the display data buffer below which the Sprite Plane stream will generate requests to memory.

2.8 Refresh Rate Hardware Control

2.8.1 RR_HW_CTL—Refresh Rate Hardware Control

RR_HW_CTL	
Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W**
Size (in bits):	**32**

This register has settings for hardware controlled switching between refresh rates. Hardware controlled refresh rate switching is enabled in the pipe configuration registers. Hardware monitors frame buffer modifications, flips, and cursor position updates, to know when to enter or exit the low power refresh rate mode. Hardware will hold a minimum number of frames in a refresh rate mode before switching to another refresh rate mode.

Bit	Description
31:16	**Reserved** Project: All Format: MBZ
15:8	**Min_High_Frames** Project: All This field specifies the minimum number of frames that must be spent in the high power refresh rate mode before allowing a switch to the low power refresh rate mode. This field is programmed to the number of frames desired minus two.
7:0	**Min_Low_Frames** Project: All This field specifies the minimum number of frames that must be spent in the low power refresh rate mode before allowing a switch to the high power refresh rate mode. This field is programmed to the number of frames desired minus two.

2.9 Backlight Control

2.9.1 BLC_PWM_CTL—Backlight PWM Control

<table>
<tr><th colspan="2">BLC_PWM_CTL</th></tr>
<tr><td colspan="2">Register Type: MMIO
Project: All
Default Value: 00000000h
Access: R/W
Size (in bits): 32</td></tr>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31</td><td><b>PWM_Enable</b>
Project: All
Default Value: 0b
This bit enables the PWM counter logic
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>Disable</td><td>PWM disabled (drives 0 always)</td><td>All</td></tr>
<tr><td>1b</td><td>Enable</td><td>PWM enabled</td><td>All</td></tr>
</table></td></tr>
<tr><td>30:29</td><td><b>PWM_Pipe_assignment</b>
Project: All
Default Value: 00b
This bit assigns PWM to a pipe. The PWM function must be disabled in order to change the value of this field.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>00b</td><td>Pipe A</td><td>Pipe A</td><td>All</td></tr>
<tr><td>01b</td><td>Pipe B</td><td>Pipe B</td><td>All</td></tr>
<tr><td>10b</td><td>Pipe C</td><td>Pipe C</td><td>All</td></tr>
<tr><td>11b</td><td>Reserved</td><td>Reserved</td><td>All</td></tr>
</table></td></tr>
<tr><td>28:27</td><td><b>Reserved</b> Project: All Format:</td></tr>
<tr><td>26</td><td><b>Phase_In_Interrupt_Status</b>
Project: All
Access: R/W Clear
Default Value: 0b
This bit will be set by hardware when a Phase-In interrupt has occurred. Software will clear this bit by writing a ‘1’, which will reset the interrupt generation.</td></tr>
<tr><td>25</td><td><b>Phase_In_Enable</b> Project: All
Setting this bit enables a PWM phase in based on the programming of the Phase In registers below. This bit clears itself when the phase in is completed.</td></tr>
<tr><td>24</td><td><b>Phase_In_Interrupt_Enable</b> Project: All
Setting this bit enables an interrupt to be generated when the PWM phase in is completed.</td></tr>
</table>

<table>
<tr><th colspan="2">BLC_PWM_CTL</th></tr>
<tr><td>23:16</td><td><b>Phase_In_time_base</b>
Project: All
Default Value: 0b
This field determines the number of VBLANK events that pass before one increment occurs.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>Invalid</td><td>Invalid</td><td>All</td></tr>
<tr><td>01h-FFh</td><td>Count</td><td>VBlank Count</td><td>All</td></tr>
</table></td></tr>
<tr><td>15:8</td><td><b>Phase_In_Count</b> Project: All
This field determines the number of increment events in this phase in. Writes to this register should only occur when hardware-phase-ins are disabled. Reads to this register can occur any time, where the value in this field indicates the number of increment events remaining to fully apply a phase-in request as hardware automatically decrements this value. A value of 0 is invalid.</td></tr>
<tr><td>7:0</td><td><b>Phase_In_Increment</b> Project: All
This field indicates the amount to adjust the PWM duty cycle register on each increment event.
This is a two’s complement number.</td></tr>
</table>

2.9.2 BLC_PWM_DATA—Backlight PWM Data

BLC_PWM_DATA	
Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W**
Size (in bits):	**32**
Bit	**Description**
31:16	**Reserved** Project: All Format:
15:0	**Backlight_Duty_Cycle** Project: All This field determines the number of time base events for the active portion of the PWM backlight control. This should never be larger than the frequency field. A value of zero will turn the backlight off. A value equal to the backlight modulation frequency field will be full on. This field gets updated when it is desired to change the intensity of the backlight, it will take affect at the end of the current PWM cycle. This value represents the active time of the PWM stream in PCH display raw clock periods multiplied by 128.

2.9.3 BLM_HIST_CTL—Image Enhancement Control

<table>
<tr><th colspan="2">BLM_HIST_CTL</th></tr>
<tr><td colspan="2">Register Type: MMIO
Project: All
Default Value: 00000000h
Access: R/W
Size (in bits): 32</td></tr>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31</td><td><b>IE_Histogram_Enable</b>
Project: All
Default Value: 0b
This bit enables the Image Enhancement histogram logic to collect data.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>Disable</td><td>Image histogram is disabled</td><td>All</td></tr>
<tr><td>1b</td><td>Enable</td><td>The image histogram is enabled. When this bit is changed from a zero to a one, histogram calculations will begin after the next VBLANK of the assigned pipe.</td><td>All</td></tr>
</table></td></tr>
<tr><td>30:29</td><td><b>IE_Pipe</b>
Project: All
Default Value: 00b
This bit assigns the IE function to a pipe. IE events will be synchronized to the VBLANK of the selected pipe. The IE function must be disabled in order to change the value of this field.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>00b</td><td>Pipe A</td><td>Pipe A</td><td>All</td></tr>
<tr><td>01b</td><td>Pipe B</td><td>Pipe B</td><td>All</td></tr>
<tr><td>10b</td><td>Pipe C</td><td>Pipe C</td><td>All</td></tr>
<tr><td>11b</td><td>Reserved</td><td>Reserved</td><td>All</td></tr>
</table></td></tr>
<tr><td>28</td><td><b>Reserved</b> Project: All Format:</td></tr>
<tr><td>27</td><td><b>IE_Modification_Table_Enable</b>
Project: All
Default Value 0b
This bit enables the Image Enhancement modification table.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>Disable</td><td>Disabled</td><td>All</td></tr>
<tr><td>1b</td><td>Enable</td><td>Enabled. When this bit is changed from a zero to a one, modifications begin after the next VBLANK of the assigned pipe.</td><td>All</td></tr>
</table></td></tr>
<tr><td>26:25</td><td><b>Reserved</b> Project: All Format:</td></tr>
</table>

<table>
<tr><th colspan="2">BLM_HIST_CTL</th></tr>
<tr><td>24</td><td><b>Histogram Mode Select</b>
Project: All
Default Value: 0b
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>YUV</td><td>YUV Luma Mode</td><td>All</td></tr>
<tr><td>1b</td><td>HSV</td><td>HSV Intensity Mode</td><td>All</td></tr>
</table></td></tr>
<tr><td>23:16</td><td><b>Sync_to_Phase_In_Count</b> Project: All
This field indicates the phase in count number on which the Image Enhancement table will be loaded if the Sync to Phase in is enabled.</td></tr>
<tr><td>15</td><td><b>IE_Table_Value_Format</b>
Project: All
Default Value: 0b 1.9
This field indicates what format is used for the image enhancement table values.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>1.9</td><td>1 integer and 9 fractional bits</td><td>All</td></tr>
<tr><td>1b</td><td>2.8</td><td>2 integer and 8 fractional bits</td><td>All</td></tr>
</table></td></tr>
<tr><td>14:13</td><td><b>Enhancement_mode</b>
Project: All
Default Value: 00b
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>00b</td><td>Direct</td><td>Direct look up mode</td><td>All</td></tr>
<tr><td>01b</td><td>Additive</td><td>Additive mode</td><td>All</td></tr>
<tr><td>10b</td><td>Multiplicative</td><td>Multiplicative mode</td><td>All</td></tr>
<tr><td>11b</td><td>Reserved</td><td>Reserved</td><td>All</td></tr>
</table></td></tr>
<tr><td>12</td><td><b>Sync_to_Phase_In</b> Project: All
Setting this bit enables the double buffered registers to be loaded on the phase in count value specified instead of the next vblank.</td></tr>
<tr><td>11</td><td><b>Bin_Register_Function_Select</b>
Project: All
Default Value: 0b
This field indicates what data is being written to or read from the bin data register.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>TC</td><td>Threshold Count. A read from the bin data register returns that bin’s threshold value from the most recent vblank load event (guardband threshold trip). Valid range for the Bin Index is 0 to 31.</td><td>All</td></tr>
<tr><td>1b</td><td>IE</td><td>Image Enhancement Value. Valid range for the Bin Index is 0 to 32</td><td>All</td></tr>
</table></td></tr>
<tr><td>10:7</td><td><b>Reserved</b> Project: All Format:</td></tr>
</table>

BLM_HIST_CTL	
6:0	**Bin_Register_Index** Project: All This field indicates the bin number whose data can be accessed through the bin data register. This value is automatically incremented by a read or a write to the bin data register if the busy bit is not set.

2.9.4 BLM_HIST_BIN—Image Enhancement Bin Data

BLM_HIST_BIN	
Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W**
Size (in bits):	**32**
Double Buffer Update Point:	**Next vblank if in normal mode, or on phase in Sync event frame if it is enabled**
Access to this address are steered to the correct register by programming the Bin Register Function Select and the Bin Register Index.	
Bit	**Description**
31	**Busy_Bit** Project: All If BLM_HIST_CTL:Bin Register Function Select = 0 (Threshold Count) This is a read only bit. If set, the engine is busy and the rest of the register is undefined. If clear, the register contains valid data. Else (Image Enhancement) This bit is reserved.
30:22	**Reserved** Project: All Format:
21:0	**Bin_Count_or_Correction_Factor** Project: All If the BLM_HIST_CTL:Bin Register Function Select = 0 (Threshold Count) { Bits 21:0 are read only bits. They indicate the total number of pixels in this bin, value is updated at the start of each vblank. } Else (Image Enhancement) { Bits 21:10 are reserved. Bits 9:0 are read/write. The program the correction value for this bin. Writes to this register are double buffered on the next vblank if in normal mode, or on the phase in Sync event frame if it is enabled. The value written here is the 10bit corrected channel value for the lowest point of the bin. }

2.9.5 BLM_HIST_GUARD—Histogram Threshold Guardband

<table>
<tr><th colspan="2">BLM_HIST_GUARD</th></tr>
<tr><td colspan="2">Register Type: MMIO
Project: All
Default Value: 00000000h
Access: R/W
Size (in bits): 32
Double Buffer Update Point: Start of vertical blank</td></tr>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31</td><td><b>Histogram_Interrupt_enable</b>
Project: All
Default Value: 0b
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>Disable</td><td>Disabled</td><td>All</td></tr>
<tr><td>1b</td><td>Enable</td><td>This generates a histogram interrupt once a Histogram event occurs. Software must always program 1.</td><td>All</td></tr>
</table></td></tr>
<tr><td>30</td><td><b>Histogram_Event_status</b>
Project: All
Access: R/W Clear
Default Value: 0b
When a Histogram event has occurred, this will get set by the hardware. For any more Histogram events to occur, the software needs to clear this bit by writing a '1'. The default state for this bit is '0'
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>Not Occurred</td><td>Histogram event has not occurred</td><td>All</td></tr>
<tr><td>1b</td><td>Occured</td><td>Histogram event has occurred</td><td>All</td></tr>
</table></td></tr>
<tr><td>29:22</td><td><b>Guardband_Interrupt_Delay</b> Project: All
An interrupt is always generated after this many consecutive frames of the guardband threshold being surpassed. This value is double buffered on start of vblank. A value of 0 is invalid.</td></tr>
<tr><td>21:0</td><td><b>Threshold_Guardband</b> Project: All
This value is used to determine the guardband for the threshold interrupt generation. This single value is used for all the segments. This value is double buffered on start of vblank</td></tr>
</table>

2.10 Motion Blur Mitigation

These registers are use to control the MBM logic. Before enabling MBM, software should have identically programmed source size, CSC, and gamma for the two pipes being used in MBM. Additionally, before executing a flip on the selected pipe (which generates the MBM frame) software should have loaded the address of the reference frame into the other pipe plane(s), this can be done by MMIO or by a flip command. The other pipe does not need to have it's panel fitter or FDI enabled or anything else down the pipe from MBM.

2.10.1 MBM_CTRL—MBM Control

<table>
<tr><th colspan="2">MBM_CTRL</th></tr>
<tr><td colspan="2">Register Type: MMIO
Project: All
Default Value: 00000000h
Access: R/W
Size (in bits): 32</td></tr>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31</td><td><b>MBM_Enable</b>
Project: All
Default Value: 0b
This bit enables MBM logic.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>Disable</td><td>MBM is Disabled</td><td>All</td></tr>
<tr><td>1b</td><td>Enable</td><td>MBM is Enabled</td><td>All</td></tr>
</table></td></tr>
<tr><td>30:29</td><td><b>Pipe_Select</b>
Project: All
Default Value: 00b
This bit assigns MBM modification to the selected pipe. The selected pipe will fetch the current buffer. The MBM Previous Buffer Pipe Select must be set to a <u>different</u> pipe to use to fetch the previous buffer.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>00b</td><td>Pipe A</td><td>Pipe A</td><td>All</td></tr>
<tr><td>01b</td><td>Pipe B</td><td>Pipe B</td><td>All</td></tr>
<tr><td>10b</td><td>Pipe C</td><td>Pipe C</td><td>All</td></tr>
<tr><td>11b</td><td>Reserved</td><td>Reserved</td><td>All</td></tr>
</table></td></tr>
<tr><td>28:27</td><td><b>Surface_select</b>
Project: All
Default Value: 0b
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>00b</td><td>None</td><td>None</td><td>All</td></tr>
<tr><td>01b</td><td>Sprite</td><td>Sprite Only</td><td>All</td></tr>
<tr><td>10b</td><td>Primary</td><td>Primary Only</td><td>All</td></tr>
<tr><td>11b</td><td>Both</td><td>Both sprite and primary</td><td>All</td></tr>
</table></td></tr>
<tr><td>26</td><td><b>Reserved</b> Project: All Format:</td></tr>
</table>

<table>
<tr><th colspan="2">MBM_CTRL</th></tr>
<tr><td>25:24</td><td><b>Previous_Buffer_Pipe_Select</b>
Project: All
Default Value: 00b
The selected pipe will fetch the previous buffer. The MBM Pipe Select must be set to a different pipe to use to fetch the current buffer and output the MBM modified pixels.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>00b</td><td>Pipe A</td><td>Pipe A</td><td>All</td></tr>
<tr><td>01b</td><td>Pipe B</td><td>Pipe B</td><td>All</td></tr>
<tr><td>10b</td><td>Pipe C</td><td>Pipe C</td><td>All</td></tr>
<tr><td>11b</td><td>Reserved</td><td>Reserved</td><td>All</td></tr>
</table></td></tr>
<tr><td>23:16</td><td><b>Delta_Threshold</b>
Project: All
Default Value: 00000000b
If the delta value between the current and previous component values exceed this threshold a compensated value is generated. Otherwise, the current value is passed through.</td></tr>
<tr><td>15:0</td><td><b>Reserved</b> Project: All Format:</td></tr>
</table>

2.10.2 MBM_TBL_INDEX—MBM Overdrive Table Index

MBM_TBL_INDEX

Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W**
Size (in bits):	**32**

These are the indexes used to access overdrive values used as the lookup table entries for MBM. Correction table factors are stored x-major, groups of input values go together, ascending from row entry zero. The first row is internally hard coded to 0, so the first address actually corresponds to the second row of the table. The last row is internally hard coded to 256, so only 7 rows total are programmable.

Overdrive table indexes:

	0	**31**	**63**	**95**	**127**	**159**	**191**	**223**	**255**
0	0	0	0	0	0	0	0	0	0
31	00h	01h	02h	03h	04h	05h	06h	07h	08h
63	09h	0Ah	0Bh	0Ch	0Dh	0Eh	0Fh	10h	11h
95	12h	13h	14h	15h	16h	17h	18h	19h	1Ah
127	1Bh	1Ch	1Dh	1Eh	1Fh	20h	21h	22h	23h
159	24h	25h	26h	27h	28h	29h	2Ah	2Bh	2Ch
191	2Dh	2Eh	2Fh	30h	31h	32h	33h	34h	35h
223	36h	37h	38h	39h	3Ah	3Bh	3Ch	3Dh	3Eh
255	256	256	256	256	256	256	256	256	256

Previous Pixel Value Range

Current Pixel Value Range

Index

Hard Coded Value

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31:16</td><td><b>Reserved</b> Project: All Format: MBZ</td></tr>
<tr><td>15</td><td><b>Index_Auto_Increment</b>
Project: All
Default Value: 0b
This field enables the index auto increment.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>No Increment</td><td>Do not automatically increment the index value.</td><td>All</td></tr>
<tr><td>1b</td><td>Auto Increment</td><td>Increment the index value with each read or write to the data register.</td><td>All</td></tr>
</table></td></tr>
<tr><td>14:6</td><td><b>Reserved</b> Project: All Format: MBZ</td></tr>
</table>

MBM_TBL_INDEX	
5:0	**Index_Value** Project: All Range 0..62 This field indicates the data location to be accessed through the data register. This value can be automatically incremented by a read or a write to the data register if the index auto increment bit is set. When automatically incrementing, the current automatically calculated index value can be read here. When automatically incrementing, the index will roll over to 0 after reaching the end of the allowed range.

2.10.3 MBM_TBL_DATA—MBM Overdrive Table Data

MBM_TBL_DATA	
Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W**
Size (in bits):	**32**
These are the overdrive values used as the lookup table entries for MBM. The MBM Overdrive Index Value indicates the MBM overdrive table location to be accessed through this register.	
Bit	**Description**
31:24	**Reserved** Project: All Format: MBZ
23:16	**Red_MBM_overdrive_value** Project: All Format: Specifies the overdrive value for the red channel.
15:8	**Green_MBM_overdrive_value** Project: All Format: Specifies the overdrive value for the green channel.
7:0	**Blue_MBM_overdrive_value** Project: All Format: Specifies the overdrive value for the blue channel.

2.11 Color Space Conversion

These registers contain the coefficients of the pipe color space converter. The high color channel is the most significant bits of the color. The low color channel is the least significant bits of the color. The medium color channel is the bits between high and low. For example: In RGB modes Red is in the High channel, Green in Medium, and Blue in Low. In YUV modes, U is in the High channel, Y in Medium, and V in Low.

The color space conversion registers are double buffered and are updated on the start of vertical blank following a write to the CSC Mode register for the respective pipe.
The matrix equations are as follows:

- OutputHigh = (CoefficientRU * InputHigh) + (CoefficientGU * InputMedium) + (CoefficientBU * InputLow)
- OutputMedium = (CoefficientRY * InputHigh) + (CoefficientGY * InputMedium) + (CoefficientBY * InputLow)
- OutputLow = (CoefficientRV * InputHigh) + (CoefficientGV * InputMedium) + (CoefficientBV * InputLow)

Example programming for RGB to YUV is in the following table:

- The input is RGB on high, medium, and low channels respectively.
- The output is VYU on high, medium, and low channels respectively.
- Program CSC_MODE to put gamma before CSC.
- Program the CSC Post-Offsets to +1/2, +1/16, and +1/2 for high, medium, and low channels respectively.
- The coefficients and pre and post offsets can be scaled if desired.

	Bt.601		Bt.709	
	Value	Program	Value	Program
RU	0.2990	0x1990	0.21260	0x2D98
GU	0.5870	0x0968	0.71520	0x0B70
BU	0.1140	0x3E98	0.07220	0x3940
RV	-0.1687	0xAAC8	-0.11460	0xBEA8
GV	-0.3313	0x9A98	-0.38540	0x9C58
BV	0.5000	0x0800	0.50000	0x0800
RY	0.5000	0x0800	0.50000	0x0800
GY	-0.4187	0x9D68	-0.45420	0x9E88
BY	-0.0813	0xBA68	-0.04580	0xB5E0

Example programming for YUV to RGB is in the following table:

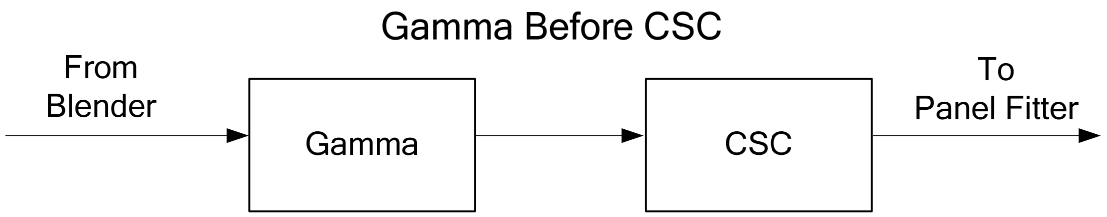
- The input is VYU on high, medium, and low channels respectively.
- The output is RGB on high, medium, and low channels respectively.
- Program CSC_MODE to put gamma after CSC.
- Program the CSC Pre-Offsets to -1/2, -1/16, and -1/2 for high, medium, and low channels respectively.
- The coefficients and pre and post offsets can be scaled if desired.

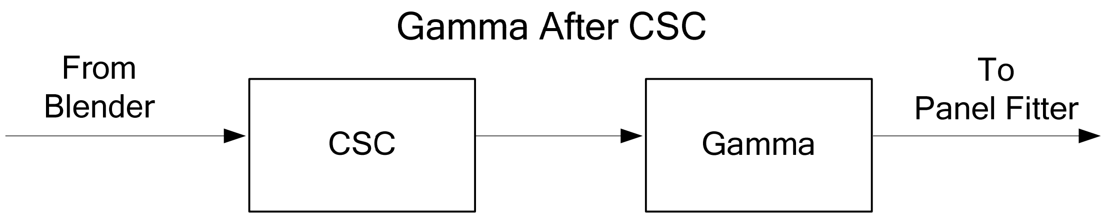
	Bt.601 Reverse		Bt.709 Reverse	
	Value	Program	Value	Program
GY	1.000	0x7800	1.000	0x7800
BY	0.000	0x0000	0.000	0x0000
RY	1.371	0x7AF8	1.574	0x7C98
GU	1.000	0x7800	1.000	0x7800
BU	-0.336	0x9AC0	-0.187	0xABF8
RU	-0.698	0x8B28	-0.468	0x9EF8
GV	1.000	0x7800	1.000	0x7800
BV	1.732	0x7DD8	1.855	0x7ED8
RV	0.000	0x0000	0.000	0x0000

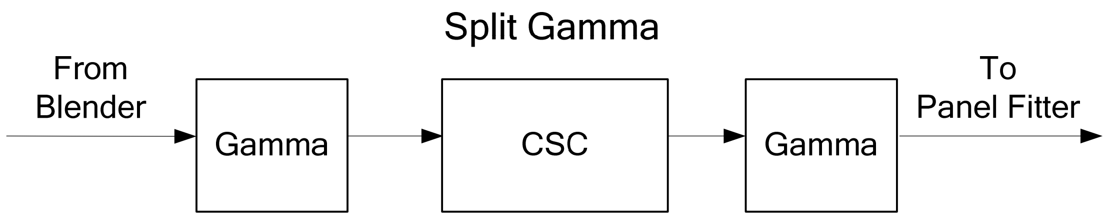
The pipe gamma and color space conversion blocks can be placed in three different arrangements:

- Gamma before CSC, selected through the CSC Mode register. This is mostly used for RGB to YUV conversion.
- Gamma after CSC, selected through the CSC Mode register. This is mostly used for YUV to RGB conversion or linear RGB to RGB conversion. This mode can be used with pipe color gamut enhancement.
- Split gamma, selected through the Pipe Config register. This is mostly used for RGB to RGB conversion. This mode can be used with pipe color gamut enhancement. In this mode, the pipe gamma enable per plane will control whether a plane will go through both gamma blocks. It is not possible to send a plane through one gamma block and not the other.

In either arrangement, the final output of the pipe gamma and CSC and gamut enhancement logic is clamped to fit in the 0 to 1.0 range before going to the ports.







2.11.1 CSC_COEFF—CSC Coefficients

<table>
<tr><th colspan="2">CSC COEFFICIENT FORMAT</th></tr>
<tr><td colspan="2"><b>Project:</b> All
<b>Default Value:</b> 0000h
<b>Size (in bits):</b> 16</td></tr>
<tr><td colspan="2">Coefficients for the CSC are stored in sign-exponent-mantissa format. Two CSC coefficients are stored in each dword, the table below show the data packing in each dword.</td></tr>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>15</td><td><b>Sign</b>
Project: All
<table><tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr><tr><td>0b</td><td>Positive</td><td>Positive</td><td>All</td></tr><tr><td>1b</td><td>Negative</td><td>Negative</td><td>All</td></tr></table></td></tr>
</table>

CSC COEFFICIENT FORMAT

<table>
<tr><td>14:12</td><td><b>Exponent_bits</b>
Project: All
Represented as 2^{-n}
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>110b</td><td>4</td><td>4 or mantissa is bb.bbbbbbb</td><td>All</td></tr>
<tr><td>111b</td><td>2</td><td>2 or mantissa is b.bbbbbbbb</td><td>All</td></tr>
<tr><td>000b</td><td>1</td><td>1 or mantissa is 0.bbbbbbbbb</td><td>All</td></tr>
<tr><td>001b</td><td>0.5</td><td>0.5 or mantissa is 0.0bbbbbbbbb</td><td>All</td></tr>
<tr><td>010b</td><td>0.25</td><td>0.25 or mantissa is 0.00bbbbbbbbb</td><td>All</td></tr>
<tr><td>011b</td><td>0.125</td><td>0.125 or mantissa is 0.000bbbbbbbbb</td><td>All</td></tr>
<tr><td>Others</td><td>Reserved</td><td>Reserved</td><td>All</td></tr>
</table></td></tr>
<tr><td>11:3</td><td><b>Mantissa</b> Project: All</td></tr>
<tr><td>2:0</td><td><b>Reserved</b> Project: All Format:</td></tr>
</table>

CSC_COEFF

Register Type: **MMIO**
Project: **All**
Default Value: **00000000h**
Access: **R/W**
Size (in bits): **6x32**
Double Buffer Update Point: **Start of vertical blank after armed**

Double Buffer Armed By: **Write to CSC_MODE**

DWord	Bit	Description			
0	31:16	**RY**	Project: All	Format:	CSC COEFFICIENT FORMAT
	15:0	**GY**	Project: All	Format:	CSC COEFFICIENT FORMAT
1	31:16	**BY**	Project: All	Format:	CSC COEFFICIENT FORMAT
	15:0	**Reserved**	Project: All	Format:	MBZ
2	31:16	**RU**	Project: All	Format:	CSC COEFFICIENT FORMAT
	15:0	**GU**	Project: All	Format:	CSC COEFFICIENT FORMAT
3	31:16	**BU**	Project: All	Format:	CSC COEFFICIENT FORMAT
	15:0	**Reserved**	Project: All	Format:	MBZ
4	31:16	**RV**	Project: All	Format:	CSC COEFFICIENT FORMAT
	15:0	**GV**	Project: All	Format:	CSC COEFFICIENT FORMAT

CSC_COEFF						
5	31:16	**BV**	Project:	All	Format:	CSC COEFFICIENT FORMAT
	15:0	**Reserved**	Project:	All	Format:	MBZ

2.11.2 CSC_MODE—CSC Mode

CSC_MODE

Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W**
Size (in bits):	**32**
Double Buffer Update Point:	**Start of vertical blank**

Writes to this register arm CSC registers for this pipe

Bit	Description
31:2	**Reserved** Project: All Format:
1	**CSC_Position** Project: All Default Value: 0b Selects the CSC position in the pipe. This is ignored when split gamma mode is selected in the pipe config register.
0	**Reserved** Project: All Format: MBZ

Value	Name	Description	Project
0b	CSC After	CSC is after gamma	All
1b	CSC Before	CSC is before gamma	All

2.11.3 CSC_PREOFF—CSC Pre-Offsets

CSC_PREOFF

Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W**
Size (in bits):	**3x32**
Double Buffer Update Point:	**Start of vertical blank after armed**
Double Buffer Armed By:	**Write to CSC_MODE**

The pre-offset is intended to remove an offset from 0 on the Y or RGB channels and to convert UV channels from excess 0.5 to 2's complement as they enter pipe color space conversion (CSC).

DWord	Bit	Description
0	31:13	**Reserved** Project: All Format: MBZ
	12:0	**PreCSC_High_Offset** Project: All This 2's complement value is used to give an offset to the high color channel as it enters CSC logic. The value is a 2's complement fraction allowing offsets between -1 and +1 (exclusive).
1	31:13	**Reserved** Project: All Format: MBZ
	12:0	**PreCSC_Medium_Offset** Project: All This 2's complement value is used to give an offset to the medium color channel as it enters CSC logic. The value is a 2's complement fraction allowing offsets between -1 and +1 (exclusive).
2	31:13	**Reserved** Project: All Format: MBZ
	12:0	**PreCSC_Low_Offset** Project: All This 2's complement value is used to give an offset to the low color channel as it enters CSC logic. The value is a 2's complement fraction allowing offsets between -1 and +1 (exclusive).

2.11.4 CSC_POSTOFF—CSC Post-Offsets

CSC_POSTOFF

Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W**
Size (in bits):	**3x32**
Double Buffer Update Point:	**Start of vertical blank after armed**
Double Buffer Armed By:	**Write to CSC_MODE**

The post-offset is intended to add an offset from 0 on the Y or RGB channels and to convert UV channels from 2's complement to excess 0.5 as they exit pipe color space conversion (CSC).

DWord	Bit	Description
0	31:13	**Reserved** Project: All Format: MBZ
	12:0	**PostCSC_High_Offset** Project: All This 2's complement value is used to give an offset to the high color channel as it exits CSC logic. The value is a 2's complement fraction allowing offsets between -1 and +1 (exclusive).
1	31:13	**Reserved** Project: All Format: MBZ
	12:0	**PostCSC_Medium_Offset** Project: All This 2's complement value is used to give an offset to the medium color channel as it exits CSC logic. The value is a 2's complement fraction allowing offsets between -1 and +1 (exclusive).
2	31:13	**Reserved** Project: All Format: MBZ
	12:0	**PostCSC_Low__Offset** Project: All This 2's complement value is used to give an offset to the low color channel as it exits CSC logic. The value is a 2's complement fraction allowing offsets between -1 and +1 (exclusive).

2.12 Pipe Palette and Gamma

The display palette provides a means to correct the gamma of an image stored in a frame buffer to match the gamma of the monitor or presentation device. Additionally, the display palette provide a method for converting indexed data values to color values for VGA and 8-bpp indexed display modes. The display palette is located after the plane blender. Using the individual plane gamma enables, the blended pixels can go through or bypass the palette on a pixel by pixel basis.

The display palette can be accessed through multiple methods and operate in one of four different modes.

8 bit legacy palette/gamma mode:

> This provides a palette mode for indexed pixel data formats (VGA and primary plane 8 bpp) and gamma correction for legacy programming requirements.
>
> All input values are clamped to the 0.0 to 1.0 range before the palette/gamma calculation. It is not recommended to use legacy palette mode with extended range formats.

For input values greater than or equal to 0 and less than 1.0, the input value is used to directly lookup the result value from one of the 256 palette/gamma entries. The 256 entries are stored in the legacy palette with 8 bits per color in a 0.8 format with 0 integer and 8 fractional bits.

The legacy palette is programmable through both MMIO and VGA I/O registers. Through VGA I/O, the palette can look as though there are only 6 bits per color component, depending on programming of other VGA I/O registers.

10 bit gamma mode:

This provides the highest quality gamma for pixel data formats of 30 bits per pixel or less.

All input values are clamped to the -3.0 to 3.0 range before the gamma calculation.

For input values greater than or equal to 0 and less than 1.0, the input value is used to directly lookup the result value from one of the first 1024 gamma entries. The first 1024 entries are stored in the precision palette with 10 bits per color in a 0.10 format with 0 integer and 10 fractional bits.

For input values greater than or equal to 1.0 and less than or equal to 3.0, the input value is used to linearly interpolate between the 1024th and 1025th gamma entries to create the result value. The 1025th entry is stored in the PAL_EXT_GC_MAX register with 19 bits per color in a 3.16 format with 3 integer and 16 fractional bits.

For negative input values, gamma is mirrored along the X-axis, giving the same result as positive input values, except for a negative sign. When gamma input may be negative, the first gamma point should be programmed to a value of 0.0 in order to have a symmetric mirroring.

Split gamma mode:

Split gamma mode is composed of two gamma functions. The first gamma is before pipe color space conversion (CSC) and the second is after CSC. This split gamma mode permits remapping to linear gamma, then color space conversion, then mapping to monitor gamma. This provides the highest quality pipe color space conversion and gamma correction for inputs with non-linear gamma.

First gamma (before CSC):

All input values are clamped to the -3.0 to 3.0 range before the gamma calculation.

For input values greater than or equal to 0 and less than 1.0, the input value is used to directly lookup the result value from one of the first 512 gamma entries. The first 512 entries are stored in the precision palette even indexes with 10 bits per color in a 0.10 format with 0 integer and 10 fractional bits.

For input values greater than or equal to 1.0 and less than or equal to 3.0, the input value is used to linearly interpolate between the 512th and 513th gamma entries to create the result value. The 513th entry is stored in the PAL_EXT_GC_MAX register with 19 bits per color in a 3.16 format with 3 integer and 16 fractional bits.

For negative input values, gamma is mirrored along the X-axis, giving the same result as positive input values, except for a negative sign. When gamma input may be negative, the first gamma point should be programmed to a value of 0.0 in order to have a symmetric mirroring.

Second gamma (after CSC):

All input values are clamped to the 0.0 to 1.0 range before the gamma calculation.

For input values greater than or equal to 0 and less than 1.0, the input value is used to directly lookup the result value from one of the first 512 gamma entries. The first 512 entries are stored in the precision palette odd indexes with 10 bits per color in a 0.10 format with 0 integer and 10 fractional bits.

12 bit interpolated gamma mode:

This provides the highest quality gamma for pixel data formats greater than 30 bits per pixel.

The gamma correction curve is represented by specifying a set of gamma entry reference points spaced equally along the curve for values between -1 and 1. For extended values there is an extended gamma entry reference point at the maximum alowed input value.

All input values are clamped to the -3.0 to 3.0 range before the gamma calculation.

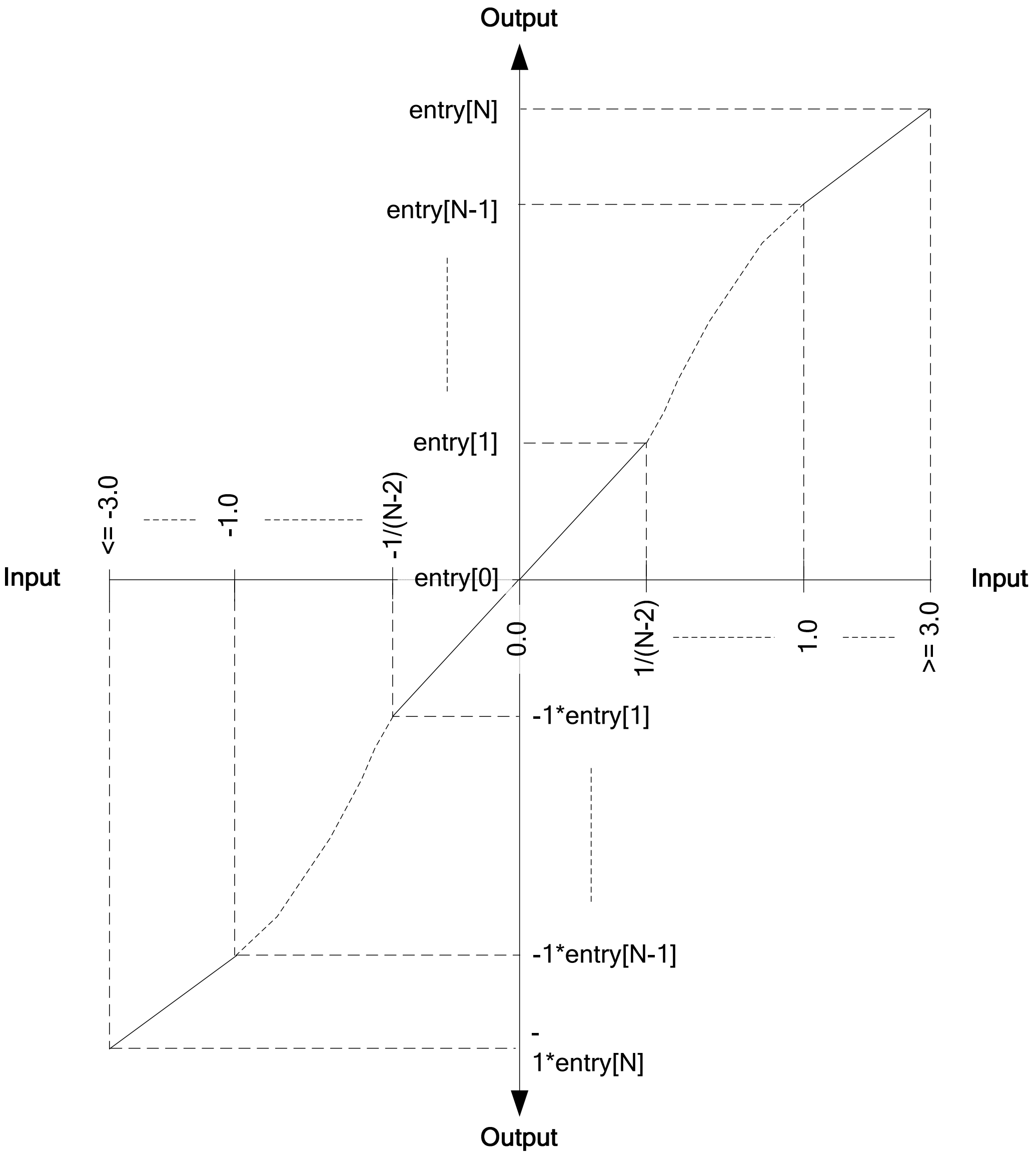
For input values greater than or equal to 0 and less than 1.0, the input value is used to linearly interpolate between two adjacent points of the first 513 gamma entries to create the result value. The first 512 entries are stored in the precision palette with 16 bits per color in a 0.16 format with 0 integer and 16 fractional bits (upper 10 bits in odd indexes, lower 6 bits in even indexes). The 513th entry is stored in the PAL_GC_MAX register with 17 bits per color in a 1.16 format with 1 integer and 16 fractional bits.

For input values greater than or equal to 1.0 and less than or equal to 3.0, the input value is used to linearly interpolate between the 513th and 514th gamma entries to create the result value. The 514th entry is stored in the PAL_EXT_GC_MAX register with 19 bits per color in a 3.16 format with 3 integer and 16 fractional bits.

For negative input values, gamma is mirrored along the X-axis, giving the same result as positive input values, except for a negative sign. When gamma input may be negative, the first gamma point should be programmed to a value of 0.0 in order to have a symmetric mirroring.

To program the gamma correction entries, calculate the desired gamma curve for inputs from 0 to 3.0. The curve must be flat or increasing, never decreasing. For inputs of 0 to 1.0, multiply the input value by 512 to find the gamma entry number, then store the desired gamma result in that entry. For inputs greater than 1.0 and less than or equal to 3.0, store the result for an input of 3.0 in the 514th gamma entry.

Example Pipe Gamma Correction Curve



2.12.1 PAL_LGC—Legacy Palette

PAL_LGC

Register Type: MMIO
Project: All
Default Value: UUUUUUUUh
Access: R/W (DWORD access only, no byte access)

DWord	Bit	Description
0..255	31:24	**Reserved** Project: All Format:
	23:16	**Red_Palette_Entry** Project: All Format:
	15:8	**Green_Palette_Entry** Project: All Format:
	7:0	**Blue_Palette_Entry** Project: All Format:

2.12.2 PAL_PREC_INDEX— Precision Palette Index

PAL_PREC_INDEX

Register Type: MMIO
Project: All
Default Value: 00000000h
Access: R/W
Size (in bits): 32

This index controls access to the array of precision palette data values.

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31:16</td><td><b>Reserved</b> Project: All Format: MBZ</td></tr>
<tr><td>15</td><td><b>Index_Auto_Increment</b>
Project: All
Default Value: 0b
This field enables the index auto increment.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>No Increment</td><td>Do not automatically increment the index value.</td><td>All</td></tr>
<tr><td>1b</td><td>Auto Increment</td><td>Increment the index value with each read or write to the data register.</td><td>All</td></tr>
</table></td></tr>
<tr><td>14:10</td><td><b>Reserved</b> Project: All Format: MBZ</td></tr>
</table>

PAL_PREC_INDEX	
9:0	**Index_Value** Project: All Range 0..1023 This field indicates the data location to be accessed through the data register. This value can be automatically incremented by a read or a write to the data register if the index auto increment bit is set. When automatically incrementing, the current automatically calculated index value can be read here. When automatically incrementing, the index will roll over to 0 after reaching the end of the allowed range.

2.12.3 PAL_PREC_DATA— Precision Palette Data

PAL_PREC_DATA	
Register Type:	**MMIO**
Project:	**All**
Default Value:	**UUUUUUUUh**
Access:	**R/W (DWORD access only, no byte access)**
Size (in bits):	**32**
These are the precision palette entries used for the 10 bpc, split, and 12 bpc gamma. The Precision Palette Index Value indicates the precision palette location to be accessed through this register	
Bit	**Description**
31:30	**Reserved** Project: All Format:
29:20	**Red_Precision_Palette_Entry** Project: All Format: For 10 bpc, program with the red 10 bit palette entry fraction value. For 12 bpc gamma odd indexes, program with the upper 10 bits of the red palette entry fraction value. For 12 bpc gamma even indexes, program the MSbs with the lower 6 bits of the red palette entry fraction value, then program all 0s in the LSbs. For split gamma even indexes, program with the first gamma (before CSC) red 10 bit palette entry fraction value. For split gamma odd indexes, program with the second gamma (after CSC) red 10 bit palette entry fraction value.
19:10	**Green_Precision_Palette_Entry** Project: All Format: For 10 bpc, program with the green 10 bit palette entry fraction value. For 12 bpc gamma odd indexes, program with the upper 10 bits of the green palette entry fraction value. For 12 bpc gamma even indexes, program the MSbs with the lower 6 bits of the green palette entry fraction value, then program all 0s in the LSbs. For split gamma even indexes, program with the first gamma (before CSC) green 10 bit palette entry fraction value. For split gamma odd indexes, program with the second gamma (after CSC) green 10 bit palette entry fraction value.

PAL_PREC_DATA	
9:0	**Blue_Precision_Palette_Entry** Project: All Format: For 10 bpc, program with the blue 10 bit palette entry fraction value. For 12 bpc gamma odd indexes, program with the upper 10 bits of the blue palette entry fraction value. For 12 bpc gamma even indexes, program the MSbs with the lower 6 bits of the blue palette entry fraction value, then program all 0s in the LSbs. For split gamma even indexes, program with the first gamma (before CSC) blue 10 bit palette entry fraction value. For split gamma odd indexes, program with the second gamma (after CSC) blue 10 bit palette entry fraction value.

2.12.4 PAL_GC_MAX—Gamma Correction Max

PAL_GC_MAX

Register Type: MMIO
Project: All
Default Value: 00010000h
Access: R/W
Size (in bits): 3x32

DWord	Bit	Description
0	31:17	**Reserved** Project: All Format: MBZ
	16:0	**Red_Max_GC_Point** Project: All The 513th entry for the red color channel of the 12 bit interpolated gamma correction. This value is represented in a 1.16 format with 1 integer and 16 fractional bits. The value should always be programmed to be less than or equal to 1.0.
1	31:17	**Reserved** Project: All Format: MBZ
	16:0	**Green_Max_GC_Point** Project: All The 513th entry for the green color channel of the 12 bit interpolated gamma correction. This value is represented in a 1.16 format with 1 integer and 16 fractional bits. The value should always be programmed to be less than or equal to 1.0.
2	31:17	**Reserved** Project: All Format: MBZ
	16:0	**Blue_Max_GC_Point** Project: All The 513th entry for the blue color channel of the 12 bit interpolated gamma correction. This value is represented in a 1.16 format with 1 integer and 16 fractional bits. The value should always be programmed to be less than or equal to 1.0.

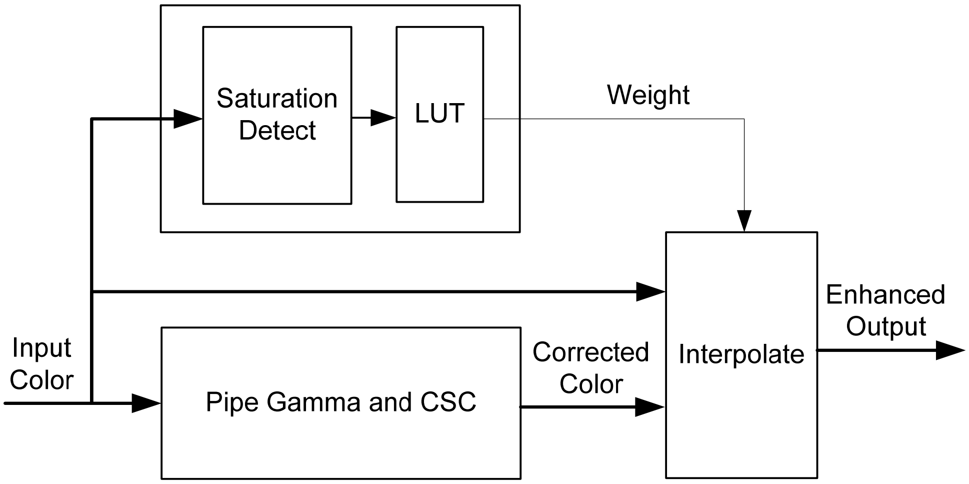
2.12.5 PAL_EXT_GC_MAX—Extended Gamma Correction Max

PAL_EXT_GC_MAX		
Register Type: MMIO **Project:** All **Default Value:** 0007FFFFh **Access:** R/W **Size (in bits):** 3x32		
DWord	**Bit**	**Description**
0	31:19	**Reserved** Project: All Format: MBZ
	18:0	**Red_Ext_Max_GC_Point** Project: All The extended point for red color channel gamma correction. This value is represented in a 3.16 format with 3 integer and 16 fractional bits. The value should always be programmed to be less than 8.0.
1	31:19	**Reserved** Project: All Format: MBZ
	18:0	**Green_Ext_Max_GC_Point** Project: All The extended point for green color channel gamma correction. This value is represented in a 3.16 format with 3 integer and 16 fractional bits. The value should always be programmed to be less than 8.0.
2	31:19	**Reserved** Project: All Format: MBZ
	18:0	**Blue_Ext_Max_GC_Point** Project: All The extended point for blue color channel gamma correction. This value is represented in a 3.16 format with 3 integer and 16 fractional bits. The value should always be programmed to be less than 8.0.

2.13 Pipe Color Gamut Enhancement

Pipe color gamut enhancement is used to enhance display of standard gamut content on wide gamut displays. It processes the color value from before and after the pipe gamma and color space correction blocks to create the color gamut enhanced output. The typical usage is to output the pipe gamma and CSC corrected color for areas of low saturated content and the input (not gamma or CSC corrected) color for areas of high saturated content. It is not recommended to use color gamut enhancement with wide gamut inputs.

The pipe Gamma and CSC must be programmed to either the split gamma mode or gamma after CSC mode when using pipe color gamut enhancement.



The saturation level of the pipe gamma and CSC input color is detected and used to index into a look up table (LUT) containing programmable weights. The saturation values are linearly distributed across the LUT indexes from the lowest index for lowest saturation to the highest index for highest saturation.

The enhanced output color is created by using the weight value to interpolate between the input color and corrected color. See the following table of weights to amount of input or corrected color used to create the enhanced output color.

Weighting of input and corrected colors

Weight from LUT	Amount of Input Color in Enhanced Output	Amount of Corrected Color in Enhanced Output
00 0000b (minimum)	0%	100%
...	...	...
00 1000b	25%	75%
...	...	...
01 0000b	50%	50%
...	...	...
01 1000b	75%	25%
...	...	...
10 0000b (maximum)	100%	0%

Example weight programming

CGE LUT Index	CGE Weight Value Decimal	CGE Weight Value Binary	CGE Weight Percent Input Color	CGE Weight Percent Corrected Color
0 (lowest saturation)	0	00 0000b	0%	100%
1	0	00 0000b	0%	100%
2	0	00 0000b	0%	100%
3	0	00 0000b	0%	100%
4	0	00 0000b	0%	100%

5	0	00 0000b	0%	100%
6	1.6	00 0010b	5%	95%
7	3.2	00 0011b	10%	90%
8	4.8	00 0101b	15%	85%
9	6.4	00 0110b	20%	80%
10	8.64	00 1001b	27%	73%
11	12.8	00 1101b	40%	60%
12	19.2	01 0011b	60%	40%
13	25.6	01 1010b	80%	20%
14	28.8	01 1101b	90%	10%
15	32	10 0000b	100%	0%
16 (highest saturation)	32	10 0000b	100%	0%

2.13.1 CGE_CTRL—Color Gamut Enhancement Control

CGE_CTRL

Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W**
Size (in bits):	**32**
Double Buffer Update Point:	**Start of vertical blank**

Bit	Description
31	**CGE_Enable** Project: All Default Value: 0b This bit enables the Color Gamut Enhancement logic.
30:0	**Reserved** Project: All Format: MBZ

Value	Name	Description	Project
0b	Disable	Disable CGE	All
1b	Enable	Enable CGE	All

2.13.2 CGE_WEIGHT—Color Gamut Enhancement Weight

CGE_WEIGHT

Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W**
Size (in bits):	**4x32**

These are the weights contained in the lookup up table (LUT) used in pipe color gamut enhancement.

LUT index 0 contains the weight for the least saturated colors, and LUT index 16 contains the weight for the most saturated colors.

Weight values can range from 00000b (100% of the enhanced output color is from the pipe gamma and CSC output corrected color) to 100000b (100% of the enhanced output color is from the pipe gamma and CSC input color).

The weight values should only be changed while color gamut enhancement is disabled, otherwise screen artifacts may show temporarily.

DWord	Bit	Description
0	31:30	**Reserved** Project: All Format: MBZ
	29:24	**CGE_Weight_Index_3** Project: All This is the weight value for this color gamut enhancement LUT index.
	23:22	**Reserved** Project: All Format: MBZ
	21:16	**CGE_Weight_Index_2** Project: All This is the weight value for this color gamut enhancement LUT index.
	15:14	**Reserved** Project: All Format: MBZ
	13:8	**CGE_Weight_Index_1** Project: All This is the weight value for this color gamut enhancement LUT index.
	7:6	**Reserved** Project: All Format: MBZ
	5:0	**CGE_Weight_Index_0** Project: All This is the weight value for this color gamut enhancement LUT index.
1	31:30	**Reserved** Project: All Format: MBZ
	29:24	**CGE_Weight_Index_7** Project: All This is the weight value for this color gamut enhancement LUT index.
	23:22	**Reserved** Project: All Format: MBZ
	21:16	**CGE_Weight_Index_6** Project: All This is the weight value for this color gamut enhancement LUT index.
	15:14	**Reserved** Project: All Format: MBZ
	13:8	**CGE_Weight_Index_5** Project: All This is the weight value for this color gamut enhancement LUT index.
	7:6	**Reserved** Project: All Format: MBZ

CGE_WEIGHT		
	5:0	**CGE_Weight_Index_4** Project: All This is the weight value for this color gamut enhancement LUT index.
2	31:30	**Reserved** Project: All Format: MBZ
	29:24	**CGE_Weight_Index_11** Project: All This is the weight value for this color gamut enhancement LUT index.
	23:22	**Reserved** Project: All Format: MBZ
	21:16	**CGE_Weight_Index_10** Project: All This is the weight value for this color gamut enhancement LUT index.
	15:14	**Reserved** Project: All Format: MBZ
	13:8	**CGE_Weight_Index_9** Project: All This is the weight value for this color gamut enhancement LUT index.
	7:6	**Reserved** Project: All Format: MBZ
	5:0	**CGE_Weight_Index_8** Project: All This is the weight value for this color gamut enhancement LUT index.
3	31:30	**Reserved** Project: All Format: MBZ
	29:24	**CGE_Weight_Index_15** Project: All This is the weight value for this color gamut enhancement LUT index.
	23:22	**Reserved** Project: All Format: MBZ
	21:16	**CGE_Weight_Index_14** Project: All This is the weight value for this color gamut enhancement LUT index.
	15:14	**Reserved** Project: All Format: MBZ
	13:8	**CGE_Weight_Index_13** Project: All This is the weight value for this color gamut enhancement LUT index.
	7:6	**Reserved** Project: All Format: MBZ
	5:0	**CGE_Weight_Index_12** Project: All This is the weight value for this color gamut enhancement LUT index.
4	31:6	**Reserved** Project: All Format: MBZ
	5:0	**CGE_Weight_Index_16** Project: All This is the weight value for this color gamut enhancement LUT index.

2.14 Software Flags

2.14.1 SWF—Software Flags

SWF		
Register Type: MMIO **Project:** All **Default Value:** 00000000h **Access:** R/W **Size (in bits):** 36x32		
These registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these registers is defined by the software architecture.		
DWord	**Bit**	**Description**
0..35	31:0	**Software_Flags** Project: All

2.14.2 GTSCRATCH—GT Scratchpad

GTSCRATCH		
Register Type: MMIO **Project:** All **Default Value:** 00000000h **Access:** R/W **Size (in bits):** 8x32		
These registers are used as scratch pad data storage space and have no direct effect on hardware operation. The use of these registers is defined by the software architecture.		
DWord	**Bit**	**Description**
0..7	31:0	**GT_Scratchpad** Project: All

3. North Display Engine Pipe and Port Controls

3.1 Pipe Timing

3.1.1 HTOTAL—Horizontal Total

HTOTAL	
Register Type: MMIO **Project:** All **Default Value:** 00000000h **Access:** R/W **Size (in bits):** 32	
Bit	**Description**
31:29	**Reserved** Project: All Format: MBZ
28:16	**Horizontal_Total** Project: All This field specifies Horizontal Total size. This should be equal to the sum of the horizontal active and the horizontal blank sizes. This field is programmed to the number of pixels desired minus one. The number of pixels (before the minus one) needs to be a multiple of two when driving the LVDS port in two channel mode. This register must always be programmed to the same value as the Horizontal Blank End.
15:12	**Reserved** Project: All Format: MBZ
11:0	**Horizontal_Active** Project: All This field specifies Horizontal Active Display size. Note that the first horizontal active display pixel is considered pixel number 0. This field is programmed to the number of pixels desired minus one. The number of pixels (before the minus one) needs to be a multiple of two when driving the LVDS port in two channel mode. The minimum horizontal active display size is 64 pixels. This register must always be programmed to the same value as the Horizontal Blank Start.

3.1.2 HBLANK—Horizontal Blank

HBLANK	
Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W**
Size (in bits):	**32**
Bit	**Description**
31:29	**Reserved** Project: All Format:
28:16	**Horizontal_Blank_End** Project: All This field specifies Horizontal Blank End position relative to the horizontal active display start. The number of pixels within horizontal blank needs to be a multiple of two when driving the LVDS port in two channel mode. The minimum horizontal blank size is 32 pixels. This register must always be programmed to the same value as the Horizontal Total.
15:13	**Reserved** Project: All Format:
12:0	**Horizontal_Blank_Start** Project: All This field specifies the Horizontal Blank Start position relative to the horizontal active display start. This register must always be programmed to the same value as the Horizontal Active.

3.1.3 HSYNC—Horizontal Sync

HSYNC	
Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W**
Size (in bits):	**32**
Bit	**Description**
31:29	**Reserved** Project: All Format: MBZ
28:16	**Horizontal_Sync_End** Project: All Default Value: 0b This field specifies the Horizontal Sync End position relative to the horizontal active display start. It is programmed with HorizontalActive+FrontPorch+Sync-1 The number of pixels within horizontal sync needs to be a multiple of two when driving the LVDS port in two channel mode. This value must be greater than the horizontal sync start and less than Horizontal Total.
15:13	**Reserved** Project: All Format: MBZ

HSYNC	
12:0	**Horizontal_Sync_Start** Project: All Default Value: 0b This field specifies the Horizontal Sync Start position relative to the horizontal active display start. It is programmed with HorizontalActive+FrontPorch-1 The number of pixels from active to horizontal sync needs to be a multiple of two when driving the LVDS port in two channel mode This value must be greater than Horizontal Active.

3.1.4 VTOTAL—Vertical Total

VTOTAL	
Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W**
Size (in bits):	**32**
Bit	**Description**
31:29	**Reserved** Project: All Format:
28:16	**Vertical_Total** Project: All This field specifies Vertical Total size. This should be equal to the sum of the vertical active and the vertical blank sizes. For progressive display modes, this field is programmed to the number of lines desired minus one. For interlaced display modes, this field is programmed with the number of lines desired minus two. The vertical counter is incremented on the leading edge of the horizontal sync. For interlaced display, hardware uses this value to calculate the vertical total in each field. Note that both even and off vertical totals are supported. This register must always be programmed to the same value as the Vertical Blank End. For Content Locked Frame Rate (CLFR) modes, hardware will be automatically adjusting the vertical total plus or minus a programmable number of lines.
15:12	**Reserved** Project: All Format:
11:0	**Vertical_Active** Project: All This field specifies Vertical Active Display size. Note that the first vertical active display line is considered pixel number 0. This field is programmed to the number of lines desired minus one. When using the internal panel fitting logic, the minimum vertical active area must be seven lines. For interlaced display, hardware uses this value to calculate the vertical active in each field. This register must always be programmed to the same value as the Vertical Blank Start.

3.1.5 VBLANK—Vertical Blank

VBLANK	
Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W**
Size (in bits):	**32**

Bit	Description
31:29	**Reserved** Project: All Format:
28:16	**Vertical_Blank_End** Project: All This field specifies Vertical Blank End position relative to the vertical active display start. The minimum vertical blank size is 5 lines. For interlaced display, hardware uses this value to calculate the vertical blank end in each field. This register must always be programmed to the same value as the Vertical Total.
15:13	**Reserved** Project: All Format:
12:0	**Vertical_Blank_Start** Project: All This field specifies the Vertical Blank Start position relative to the vertical active display start. For interlaced display, hardware uses this value to calculate the vertical blank start in each field. This register must always be programmed to the same value as the Vertical Active

3.1.6 VSYNC—Vertical Sync

VSYNC	
Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W**
Size (in bits):	**32**

Bit	Description
31:29	**Reserved** Project: All Format:
28:16	**Vertical_Sync_End** Project: All This field specifies the Vertical Sync End position relative to the vertical active display start. It is programmed with VerticalActive+FrontPorch+Sync-1 For interlaced display, hardware uses this value to calculate the vertical sync start in each field. This value must be greater than the vertical sync start and less than Vertical Total.
15:13	**Reserved** Project: All Format:

VSYNC	
12:0	**Vertical_Sync_Start** Project: All This field specifies the Vertical Sync Start position relative to the vertical active display start. It is programmed with VerticalActive+FrontPorch-1 For interlaced display, hardware uses this value to calculate the vertical sync end in each field. This value must be greater than Vertical Active.

3.1.7 SRCSZ—Source Image Size

SRCSZ	
Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W**
Size (in bits):	**32**
Double Buffer Update Point:	**Start of vertical blank**

In VGA display mode, this register is ignored and the VGA size from the VGA registers is used instead.

This register may be updated through MMIO writes or through a command streamer initiated synchronous flip.

Bit	Description
31:28	**Reserved** Project: All Format: MBZ
27:16	**Horizontal_Source_Size** Project: All This field specifies Horizontal Source Size. This determines the horizontal size of the image created by the display planes. This field is programmed to the number of pixels desired minus one. This register must always be programmed to the same value as the Horizontal Active, except when panel fitting is enabled.
15:12	**Reserved** Project: All Format: MBZ
11:0	**Vertical_Source_Size** Project: All This field specifies Vertical Source Size. This determines the vertical size of the image created by the display planes. This field is programmed to the number of lines desired minus one. For interlaced display, hardware divides this number by 2 and adds any necessary half lines to get the vertical blank end for each field. This register must always be programmed to the same value as the Vertical Active, except when panel fitting is enabled.

3.1.8 VSYNCSHIFT— Vertical Sync Shift

VSYNCSHIFT	
Register Type: MMIO Project: All Default Value: 00000000h Access: R/W Size (in bits): 32	
Bit	**Description**
31:13	**Reserved** Project: All Format:
12:0	**Second_Field_VSync_Shift** Project: All This value specifies the vertical sync alignment for the start of the interlaced second field, expressed in terms of the absolute pixel number relative to the horizontal active display start. This value will only be used if the pipe is programmed to an interlaced mode. Typically, the interlaced second field vertical sync should start one pixel after the point halfway between successive horizontal syncs, so the value of this register should be programmed to: horizontal sync start - floor[horizontal total / 2] (use the actual horizontal sync start and horizontal total values and not the minus one values programmed into the registers) This vertical sync shift only occurs during the interlaced second field. In all other cases the vertical sync start position is aligned with horizontal sync start.

3.2 Pipe M/N Values

These values are used for the embedded FDI.

When switching between two refresh rates, both the M1/N1 data and link values and the M2/N2 link values must be programmed. For dynamic refresh rate control, M1/N1 values are the primary values and are used for the normal M/N value setting, and M2/N2 values are the secondary values and are used for the lower power M/N value setting.

Calculation of TU, Data M, and Data N is as follows:
For modes that divide into the link frequency evenly,
Active/TU = Payload/Capacity = Data M/N = dot clock * bytes per pixel / ls_clk * number of lanes
Default value to program TU size is "111111" for TU size of 64.

Calculation of Link M and Link N is as follows:
Link M/N = dot clock / ls_clk

Restriction on clocks and number of lanes:
Number of lanes >= INT(dot clock * bytes per pixel / ls_clk)
Pcdclk * number of lanes >= dot clock * bytes per pixel

3.2.1 DATAM— Data M Value

DATAM

Register Type:	MMIO
Project:	All
Default Value:	00000000h
Access:	R/W
Size (in bits):	32
Double Buffer Update Point:	Start of vertical blank
Double Buffer Armed By:	Writing the LINKN

Bit	Description
31	**Reserved** Project: All Format: MBZ
30:25	**TU_Size** Project: All This field is the size of the transfer unit, minus one.
24	**Reserved** Project: All Format: MBZ
23:0	**Data_M_value** Project: All This field is the m value for internal use of the DDA.

3.2.2 DATAN— Data N Value

DATAN

Register Type:	MMIO
Project:	All
Default Value:	00000000h
Access:	R/W
Size (in bits):	32
Double Buffer Update Point:	Start of vertical blank
Double Buffer Armed By:	Writing the LINKN

Bit	Description
31:24	**Reserved** Project: All Format: MBZ
23:0	**Data_N_value** Project: All This field is the n value for internal use of the DDA.

3.2.3 LINKM— Link M Value

LINKM

Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W**
Size (in bits):	**32**
Double Buffer Update Point:	**Start of vertical blank**
Double Buffer Armed By:	**Writing the LINKN**

Bit	Description
31:24	**Reserved** Project: All Format: MBZ
23:0	**Link_M_value** Project: All This field is the m value for external transmission in the Main Stream Attributes.

3.2.4 LINKN— Link N Value

LINKN

Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W**
Size (in bits):	**32**
Double Buffer Update Point:	**Start of vertical blank**

Writes to this register arm M/N registers for this pipe.

Bit	Description
31:24	**Reserved** Project: All Format: MBZ
23:0	**Link_N_value** Project: All This field is the n value for external transmission in the Main Stream Attributes and VB-ID.

3.3 FDI Transmit

3.3.1 FDI_TX_CTL—FDI Tx Control

<table>
<tr><th colspan="2">FDI_TX_CTL</th></tr>
<tr><td colspan="2">Register Type: MMIO
Project: All
Default Value: 00040000h
Access: R/W
Size (in bits): 32
Double Buffer Update Point: Depends on Bit</td></tr>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31</td><td><b>FDI_Tx_Enable</b>
Project: All
Default Value: 0b
Disabling this port will put it in its lowest power state. Port enable takes place on the Vblank after being written.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>Disable</td><td>Disable and tristate the FDI Tx interface</td><td>All</td></tr>
<tr><td>1b</td><td>Enable</td><td>Enable the FDI Tx interface</td><td>All</td></tr>
</table></td></tr>
<tr><td>30:28</td><td><b>Reserved</b> Project: All Format: MBZ</td></tr>
<tr><td>27:25</td><td><b>Voltage_swing_level_set</b>
Project: All
Default Value: 000b
These bits are used for setting the voltage swing for pattern 1.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>000b</td><td>0.4V</td><td>0.4V</td><td>All</td></tr>
<tr><td>001b</td><td>0.6V</td><td>0.6V</td><td>All</td></tr>
<tr><td>010b</td><td>0.8V</td><td>0.8V</td><td>All</td></tr>
<tr><td>011b</td><td>1.2V</td><td>1.2V</td><td>All</td></tr>
<tr><td>Others</td><td>Reserved</td><td>Reserved</td><td>All</td></tr>
</table></td></tr>
</table>

FDI_TX_CTL

24:22

Preemphasis_level_set

Project: All

Default Value: 000b

These bits are used for setting link pre-emphasis for pattern 2.

Value	Name	Description	Project
000b	0dB	No pre-emphasis	All
001b	3.5dB	3.5dB pre-emphasis (1.5x)	All
010b	6dB	6dB pre-emphasis (2x)	All
011b	9.5dB	9.5dB pre-emphasis (3x)	All
Others	Reserved	Reserved	All

21:19

Port_Width_Selection

Project: All

Default Value: 000b

This bit selects the number of lanes to be enabled on the link. Port width change must be done as a part of mode set.

[DevIVB] FDI B and FDI C share lanes. FDI C maximum port width is 2 lanes. FDI B maximum port width is 4 lanes when FDI C is disabled, 2 lanes when FDI C is enabled.

Locked once port is enabled. Updates when the port is disabled then re-enabled.

Value	Name	Description	Project
000b	X1	x1 Mode	All
001b	X2	x2 Mode	All
010b	X3	x3 Mode	All
011b	X4	x4 Mode	All
Others	Reserved	Reserved	All

18

Enhanced_Framing_Enable

Project: All

Default Value: 1b Enhanced framing enabled

This bit selects enhanced framing.

Locked once port is enabled. Updates when the port is disabled then re-enabled

Value	Name	Description	Project
0b	Disabled	Enhanced framing disabled	All
1b	Enabled	Enhanced framing enabled	All

17:15

Reserved Project: All Format: MBZ

FDI_TX_CTL

14

FDI_PLL_enable

Project: All

Default Value: 0b

This bit enables the FDI PLL. Software must wait for the PLL warmup cycle before enabling the port through bit 31 of this register. This bit is ORed with the PLL enable bit from any other FDI Tx Control registers.

Value	**Name**	**Description**	**Project**
0b	Disable	FDI PLL not enabled through this FDI Tx	All
1b	Enable	FDI PLL enabled	All

13:12

Reserved Project: All Format: MBZ

11

Composite_Sync_Select

Project: All

Default Value: 0b

This bit selects between composite Sync and separate Fsync/Lsync on this port.

Value	**Name**	**Description**	**Project**
0b	Separate	Separate Fsync/Lsync	All
1b	Composite	Composite Sync	All

10

Auto_Train

Project: All

Default Value: 0b

This bit enables auto-training on this port.

Value	**Name**	**Description**	**Project**
0b	Disable	Disable auto-training	All
1b	Enable	Enable auto-training	All

FDI_TX_CTL

9:8 **Link_training_pattern_enable**

Project: All

Default Value: 00b

These bits are used for link initialization. Please note that the link must first be configured prior to sending training patterns.

When enabling the port, it must be turned on with pattern 1 enabled. When retraining a link, the port must be disabled, then re-enabled with pattern 1 enabled.

Value	Name	Description	Project
00b	P1	Pattern 1 enabled: Repetition of D10.2 characters	All
01b	P2	Pattern 2 enabled: Repetition of K28.5, D11.6, K28.5, D11.6, D10.2, D10.2, D10.2, D10.2, D10.2, D10.2. Please note that the entire pattern must complete before another pattern is sent. Scrambling initialization and disparity init commence at the end of the last iteration of pattern 2.	All
10b	Idle	Idle Pattern enabled: Transmit BS followed by VB-ID with NoVideoStream_flag set to 1, five times	All
11b	None	Link not in training: Send normal pixels	All

6 **Reserved** Project: All Format: MBZ

5 **Encryption_Port_Select**

Project: All

Default Value: 0b

This bit directs encryption to this port. When selected, the information sent on this port will be encrypted. Please note that this bit does not enable encryption on its own, but must be used in conjunction with the encryption registers.

Value	Name	Description	Project
0b	No encryption	No encryption on this port	All
1b	Select encryption	Select encryption on this port	All

4:1 **Reserved** Project: All Format: MBZ

0 **Master_enable**

Project: All

Security: Test

Default Value: 0b

Setting this bit to ‘1’ forces the timing generator to operate in master mode. Otherwise, Lsync/Fsync inputs are required for FDI port operation.

3.4 Panel Fitter

3.4.1 PF_PWR_GATE—Panel Fitter Power Gate Control

<table>
<tr><th colspan="2">PF_PWR_GATE</th></tr>
<tr><td>Register Type:</td><td>MMIO</td></tr>
<tr><td>Project:</td><td>All</td></tr>
<tr><td>Default Value:</td><td>00006453h</td></tr>
<tr><td>Access:</td><td>R/W</td></tr>
<tr><td>Size (in bits):</td><td>32</td></tr>
<tr><td>Double Buffer Update Point:</td><td>Start of vertical blank after armed</td></tr>
<tr><td>Double Buffer Armed By:</td><td>Write to PF_WIN_SZ</td></tr>
</table>

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31:16</td><td><b>Reserved</b> Project: All Format: MBZ</td></tr>
<tr><td>15:13</td><td><b>LATE_SIGNAL_SEQUENCE_START</b> Project: All
Start of the late signal to the first RAM bank in number of cdclks after the start of power gating sequence. Applicable for both power gating on and off conditions.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>000b</td><td>Start time 0</td><td>Start time 0</td><td>All</td></tr>
<tr><td>001b</td><td>Start time 256</td><td>Start time 256</td><td>All</td></tr>
<tr><td>010b</td><td>Start time 512</td><td>Start time 512</td><td>All</td></tr>
<tr><td>011b</td><td>Start time 768</td><td>Start time 768</td><td>All</td></tr>
<tr><td>100b</td><td>Start time 1024</td><td>Start time 1024</td><td>All</td></tr>
<tr><td>101b</td><td>Start time 1280</td><td>Start time 1280</td><td>All</td></tr>
<tr><td>110b</td><td>Start time 1536</td><td>Start time 1536</td><td>All</td></tr>
<tr><td>111b</td><td>Start time 1792</td><td>Start time 1792</td><td>All</td></tr>
</table></td></tr>
<tr><td>12</td><td><b>Reserved</b> Project: All Format: MBZ</td></tr>
</table>

PF_PWR_GATE

11:9 **MID_SIGNAL_SEQUENCE_START** Project: All

Start of the mid signal to the first RAM bank in number of cdclks after the start of power gating sequence. Applicable for both power gating on and off conditions.

Value	Name	Description	Project
000b	Start time 0	Start time 0	All
001b	Start time 256	Start time 256	All
010b	Start time 512	Start time 512	All
011b	Start time 768	Start time 768	All
100b	Start time 1024	Start time 1024	All
101b	Start time 1280	Start time 1280	All
110b	Start time 1536	Start time 1536	All
111b	Start time 1792	Start time 1792	All

8 **Reserved** Project: All Format: MBZ

7:6 **LATE_SIGNAL_DELAY** Project: All

Delay between late signals going into successive RAM banks in number of cdclks. Applicable for both power gating on and off conditions.

Value	Name	Description	Project
00b	Start time 0	Start time 0	All
01b	Start time 256	Start time 256	All
10b	Start time 512	Start time 512	All
11b	Start time 768	Start time 768	All

5 **Reserved** Project: All Format: MBZ

4:3 **MID_SIGNAL_DELAY** Project: All

Delay between mid signals going into successive RAM banks in number of cdclks. Applicable for both power gating on and off conditions.

Value	Name	Description	Project
00b	Start time 0	Start time 0	All
01b	Start time 256	Start time 256	All
10b	Start time 512	Start time 512	All
11b	Start time 768	Start time 768	All

2 **Reserved** Project: All Format: MBZ

PF_PWR_GATE	
1:0	**EARLY_SIGNAL_DELAY** Project: All Delay between early signals going into successive RAM banks in number of cdclks. Applicable for both power gating on and off conditions.

Value	Name	Description	Project
00b	Start time 0	Start time 0	All
01b	Start time 256	Start time 256	All
10b	Start time 512	Start time 512	All
11b	Start time 768	Start time 768	All

3.4.2 PF_WIN_POS—Panel Fitter Window Position

PF_WIN_POS	
Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W**
Size (in bits):	**32**
Double Buffer Update Point:	**Start of vertical blank after armed**
Double Buffer Armed By:	**Write to PF_WIN_SZ**

Bit	Description
31:29	**Reserved** Project: All Format: MBZ
28:16	**XPOS** Project: All The X coordinate in pixels of the upper left most pixel of the panel fitted display window.
15:12	**Reserved** Project: All Format: MBZ
11:0	**YPOS** Project: All The Y coordinate in lines of the upper left most pixel of the panel fitter display window. LSB must be zero for interlaced modes.

3.4.3 PF_WIN_SZ—Panel Fitter Window Size

PF_WIN_SZ

Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W**
Size (in bits):	**32**
Double Buffer Update Point:	**Start of vertical blank**

Coordinates are determined with a value of (0,0) being the upper left corner of the display device (rotation does not affect this). **Writes to this register arm PF registers on this pipe.**

Bit	Description
31:29	**Reserved** Project: All Format: MBZ
28:16	**XSIZE** Project: All The horizontal size in pixels of the desired panel fitted window.
15:12	**Reserved** Project: All Format: MBZ
11:0	**YSIZE** Project: All The vertical size in pixels of the desired panel fitted window. LSB must be zero for interlaced modes.

3.4.4 PF_CTRL—Panel Fitter Control

<table>
<tr><th colspan="2">PF_CTRL</th></tr>
<tr><td colspan="2">
<table>
<tr><td>Register Type:</td><td>MMIO</td></tr>
<tr><td>Project:</td><td>All</td></tr>
<tr><td>Default Value:</td><td>00000000h</td></tr>
<tr><td>Access:</td><td>R/W</td></tr>
<tr><td>Size (in bits):</td><td>32</td></tr>
<tr><td>Double Buffer Update Point:</td><td>Start of vertical blank after armed</td></tr>
<tr><td>Double Buffer Armed By:</td><td>Write to PF_WIN_SZ</td></tr>
</table>
When using panel fitter downscaling (pipe source size is larger than panel fitter window size) the maximum supported pixel rate will be reduced by the downscale amount.
</td></tr>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31</td><td>
<b>Enable_Pipe_Scaler</b>
Project: All
Default Value: 0b
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>Disable</td><td>Data bypasses the scaler</td><td>All</td></tr>
<tr><td>1b</td><td>Enable</td><td>The scaler is enabled</td><td>All</td></tr>
</table>
</td></tr>
<tr><td>30:29</td><td>
<b>Pipe_Select</b>
Project: All
Default Value: 00b
This bit determines which display pipe this panel fitter will connect to. Do not enable and connect more than one panel fitter to a pipe.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>00b</td><td>Pipe A</td><td>Pipe A</td><td>All</td></tr>
<tr><td>01b</td><td>Pipe B</td><td>Pipe B</td><td>All</td></tr>
<tr><td>10b</td><td>Pipe C</td><td>Pipe C</td><td>All</td></tr>
<tr><td>11b</td><td>Reserved</td><td>Reserved</td><td>All</td></tr>
</table>
</td></tr>
</table>

PF_CTRL

28 **V_FILTER_BYPASS**

Project: All
Security: Test
Default Value: 0b

Bypass the Vertical Filter

Value	Name	Description	Project
0b	Enable	Vertical Filter Enabled	All
1b	Bypass	Vertical Filter Bypassed	All

27 **VADAPT_EN** Project: All

Enables the adaptive vertical filter, intended for use with YUV data in interlace output modes only.

Adaptive vertical filter only works with 7x5 capable panel fitters. For panel fitters that are only 3x3 capable, this must not be enabled.

Value	Name	Description	Project
0b	Disable	Adaptive filtering disabled	All
1b	Enable	Adaptive filtering enabled	All

26:25 **VADAPT_MODE** Project: All

Selects adaptive vertical filter mode.

Adaptive vertical filter only works with 7x5 capable panel fitters. For panel fitters that are only 3x3 capable, this field is ignored.

Value	Name	Description	Project
00b	Least Adaptive	Least Adaptive (Recommended)	All
01b	Moderately Adaptive	Moderately Adaptive	All
10b	Reserved	Reserved	All
11b	Most Adaptive	Most Adaptive	All

24:23 **FILTER_SELECT** Project: All

Selects filter coefficients.

Programmed coefficients only work with 7x5 capable panel fitters. For panel fitters that are only 3x3 capable, this field must be programmed to select one of the hardcoded coefficient sets.

Value	Name	Description	Project
00b	Programmed	Programmed Coefficients (Recommended for 7x5 capable panel fitters, not available for 3x3 capable panel fitters)	All
01b	Hardcoded Med	Hardcoded Coefficients for Medium 3x3 Filtering	All
10b	Hardcoded Edge Enhance	Hardcoded Coefficients for Edge Enhancing 3x3 Filtering	All
11b	Hardcoded Edge Soften	Hardcoded Coefficients for Edge Softening 3x3 Filtering	All

22 **Reserved** Project: All Format: MBZ

PF_CTRL

21 **VERT3TAP**

Project: All
Security: Test
Default Value: 0b

Value	Name	Description	Project
0b	Auto	Auto-detection of 3 tap usage	All
1b	Force	Force 3 tap vertical scaling	All

20 **VERTICAL_INT_FIELD_INVERT**

Project: All
Security: Test
Default Value: 0b

Value	Name	Description	Project
0b	Field 1	Field 1 will get the phase increment (vertical initial phase) in the vertical filter for interlace.	All
1b	Field 0	Field 0 will get the phase increment (vertical initial phase) in the vertical filter for interlace.	All

19 **AUTO_SCALE_MODE**

Project: All
Security: Test
Default Value: 0b

Value	Name	Description	Project
0b	Auto	The scaler will calculate the scale factors automatically, selected fractions can be read back in the other filter control registers.	All
1b	Non-auto	The scaler will use the scaling factors written in the other filter control registers	All

18 **AUTO_SCALE_CALC** Project: All Security: Test

Access: Read Only

This read only bit will be set while the auto scale function is in progress. It indicates that the values read back from the rest of the filter control registers should be ignored.

17:0 **Reserved** Project: All Format: MBZ

3.4.5 PF_VSCALE—Panel Fitter Vertical Scale

PF_VSCALE	
Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W**
Size (in bits):	**32**
Double Buffer Update Point:	**Start of vertical blank after armed**
Double Buffer Armed By:	**Write to PF_WIN_SZ**

This register is read only in the auto-scale mode.

Bit	Description
31:18	**Reserved** Project: All Format: MBZ
17:15	**VSCALE_INT** Project: All The integer part of the vertical scale factor. VSCALE_INT = int(src height/(interlace x dest height)) where interlace = 1/2 in interlace modes, 1 in progressive modes
14:0	**VSCALE_FRAC** Project: All The fractional part of the vertical scale factor. VSCALE_FRAC = int((src height/(interlace x dest height)-VSCALE_INT)*2^^15) where interlace = 1/2 in interlace modes, 1 in progressive modes

3.4.6 PF_VSCALE_IP—Panel Fitter Vertical Scale IP

PF_VSCALE_IP	
Register Type:	MMIO
Project:	All
Default Value:	00000000h
Access:	R/W
Size (in bits):	32
Double Buffer Update Point:	Start of vertical blank after armed
Double Buffer Armed By:	Write to PF_WIN_SZ

This register is read only in the auto-scale mode. For non-auto-scale progressive scan modes this register should be programmed to all zeroes.

Bit	Description
31:18	**Reserved** Project: All Format: MBZ
17:15	**VSCALE_INT_IP** Project: All The integer portion of the initial phase of the vertical scaler for the bottom field VSCALE_INT_IP = int((source height)/(destination height))
14:0	**VSCALE_FRAC_IP** Project: All The fractional portion of the initial phase of the vertical scaler for the bottom field VSCALE_FRAC_IP = int((source height-1)/(destination height-1)-VSCALE_INT_IP)*2^^15))

3.4.7 PF_HSCALE—Panel Fitter Horizontal Scale

PF_HSCALE

Register Type:	MMIO
Project:	All
Default Value:	00000000h
Access:	R/W
Size (in bits):	32
Double Buffer Update Point:	Start of vertical blank after armed
Double Buffer Armed By:	Write to PF_WIN_SZ

This register is read only in the auto-scale mode.

Bit	Description
31:18	**Reserved** Project: All Format: MBZ
17:15	**HSCALE_INT** Project: All The integer part of the horizontal scaling factor divided by the oversampling rate. HSCALE_INT = int(src width/dest width)
14:0	**HSCALE_FRAC** Project: All The fractional part of the horizontal scaling factor divided by the oversampling rate. HSCALE_FRAC = int(((src width/dest width)-HSCALE_INT)*2^^14)

3.4.8 PF_COEF_INDEX—Panel Fitter Coefficients Index

Horizontal coefficients are accessed through the index and data registers following the mapping shown below. 17 phase of 7 taps requires 119 coefficients in 60 dwords per set. The letter represents the filter tap (D is the center tap) and the number represents the coefficient set for a phase (0-16).

Horizontal Luma/Red Coefficient Mapping			Horizontal Chroma/Green/Blue Coefficient Mapping		
Index Value	**Data Value Coefficient2**	**Data Value Coefficient1**	**Index Value**	**Data Value Coefficient2**	**Data Value Coefficient1**
00h	B0	A0	3Ch	B0	A0
01h	D0	C0	3Dh	D0	C0
02h	F0	E0	3Eh	F0	E0
03h	A1	G0	3Fh	A1	G0
04h	C1	B1	40h	C1	B1
...	...	...	...	...	...
38h	B16	A16	74h	B16	A16
39h	D16	C16	75h	D16	C16
3Ah	F16	E16	76h	F16	E16
3Bh	Reserved	G16	77h	Reserved	G16

Vertical coefficients are accessed through the index and data registers following the mapping shown below. 17 phase of 5 taps requires 85 coefficients in 43 dwords per set. The letter represents the filter tap (C is the center tap) and the number represents the coefficient set for a phase (0-16).

Vertical Luma/Red Coefficient Mapping			Vertical Chroma/Green/Blue Coefficient Mapping		
Index Value	**Data Value Coefficient2**	**Data Value Coefficient1**	**Index Value**	**Data Value Coefficient2**	**Data Value Coefficient1**
00h	B0	A0	2Bh	B0	A0
01h	D0	C0	2Ch	D0	C0
02h	A1	E0	2Dh	A1	E0
03h	C1	B1	2Eh	C1	B1
...	...	...	...	...	...
27h	B16	A16	53h	B16	A16
28h	D16	C16	54h	D16	C16
2Ah	Reserved	E16	55h	Reserved	E16

PF_COEF_INDEX

Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W**
Size (in bits):	**32**

This index controls access to the array of panel fitter coefficient data values. See the coefficient mapping tables for information on mapping of index to data values for each set of coefficients.

<table>
<tr><th colspan="2">PF_COEF_INDEX</th></tr>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31:16</td><td><b>Reserved</b> Project: All Format: MBZ</td></tr>
<tr><td>15</td><td><b>Index_Auto_Increment</b>
Project: All
Default Value: 0b
This field enables the index auto increment.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>No Increment</td><td>Do not automatically increment the index value.</td><td>All</td></tr>
<tr><td>1b</td><td>Auto Increment</td><td>Increment the index value with each read or write to the data register.</td><td>All</td></tr>
</table></td></tr>
<tr><td>14:7</td><td><b>Reserved</b> Project: All Format: MBZ</td></tr>
<tr><td>6:0</td><td><b>Index_Value</b>
Project: All
This field indicates the data location to be accessed through the data register. This value can be automatically incremented by a read or a write to the data register if the index auto increment bit is set. When automatically incrementing, the current automatically calculated index value can be read here. When automatically incrementing, the index will roll over to 0 after reaching the end of the allowed range.</td></tr>
</table>

3.4.9 PF_COEF_DATA—Panel Fitter Coefficients Data

<table>
<tr><th colspan="2">Panel Fitter Coefficient Format</th></tr>
<tr><th>Project:</th><th>All</th></tr>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>15</td><td><b>Sign</b>
Project: All
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>Positive</td><td>Positive</td><td>All</td></tr>
<tr><td>1b</td><td>Negative</td><td>Negative</td><td>All</td></tr>
</table></td></tr>
<tr><td>14</td><td><b>Reserved</b> Project: All Format: MBZ</td></tr>
</table>

Panel Fitter Coefficient Format

13:12 **Exponent**

Project: All

The meaning of the exponent bits varies for center tap or non-center tap coefficients.

Value	Name	Description	Project
00b	2 or 0.125	Center taps: 2 or mantissa is b.bbbbbbbb Non-center taps: 0.125 or mantissa is 0.000bbbbbbb	All
01b	1	1 or mantissa is 0.bbbbbbb..	All
10b	0.5	0.5 or mantissa is 0.0bbbbbbb..	All
11b	0.25	0.25 or mantissa is 0.00bbbbbbb..	All
Others	Reserved	Reserved	All

11:3 **Mantissa** Project: All

Size of the mantissa varies based on the filter, but the MSB of the mantissa is always bit 11.
Center tap coefficients use all 9 bits of mantissa.
Non-center tap coefficients use only the upper 7 bits of mantissa and the lower 2 bits are ignored.

2:0 **Reserved** Project: All Format: MBZ

Create new PF coefficient data register

PF_COEF_DATA

Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W (DWORD access only, no byte access)**
Size (in bits):	**32**

These are the coefficient values for panel fitter. The Panel Fitter Coefficients Index Value indicates the Panel Fitter Coefficients location to be accessed through this register.

Bit	Description
31:16	**Coefficient2** Project: All Format: Panel Fitter Coefficient Format Specifies the value for the second coefficient stored in this dword.
15:0	**Coefficient1** Project: All Format: Panel Fitter Coefficient Format Specifies the value for the first coefficient stored in this dword.

3.4.10 PIPE_SCANLINE—Pipe Scan Line

PIPE_SCANLINE	
Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**Read Only**
Size (in bits):	**32**

This register enables the read back of the display pipe vertical "line counter". The value increments at the leading edge of HSYNC and can be safely read any time. The value resets to line zero at the first active line of the display. In interlaced display timings, the scan line counter provides the current line in the field. One field can have a total number of lines that is one greater than the other field. Content locked display will adjust the total number of lines displayed.

Bit	Description
31	**Current_Field** Project: All Default Value: 0 Provides read back of the current field being displayed on the display pipe.
30:13	**Reserved** Project: All Format:
12:0	**Line_Counter_for_Display** Project: All Provides read back of the display pipe vertical line counter. This is an indication of the current display scan line.

Current_Field values (bit 31):

Value	Name	Description	Project
0b	Odd	First field (odd field)	All
1b	Even	Second field (even field)	All

4. North Display Engine Pipe and Plane Controls

4.1 Pipe Control

4.1.1 PIPE_SCANLINECOMP—Pipe Scan Line Compare

PIPE_SCANLINECOMP

Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W**
Size (in bits):	**32**

The scan line number register is compared with the display line value from the pipe timing generator. The result of this comparison is used to generate interrupts and render responses.

The value programmed should be desired value – 1, so for line 0, the value programmed is VTOTAL, and for line 1, the value programmed is 0. In interlaced display timings, the scan line is per field. One field can have a total number of lines that is one greater than the other field. Content locked display will adjust the total number of lines displayed.

Bit	Description
31:13	**Reserved** Project: All Format: MBZ
12:0	**Scan_Line_Number** Project: All Range 0..Vertical Total This field specifies the scan line number on which to generate scan line interrupt and render response.

4.1.2 PIPE_CONF—Pipe Configuration

<table>
<tr><th colspan="2">PIPE_CONF</th></tr>
<tr><td>Register Type:</td><td>MMIO</td></tr>
<tr><td>Project:</td><td>All</td></tr>
<tr><td>Default Value:</td><td>00000000h</td></tr>
<tr><td>Access:</td><td>R/W</td></tr>
<tr><td>Size (in bits):</td><td>32</td></tr>
<tr><td>Double Buffer Update Point:</td><td>Start of vertical blank OR pipe disabled</td></tr>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31</td><td><b>Pipe_Enable</b>
Project: All
Default Value: 0b
Setting this bit to the value of one, turns on this pipe. Turning the pipe off disables the timing generator and synchronization pulses to the display will not be maintained. Pipe timing registers must contain valid values before this bit is enabled.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>Disable</td><td>Disable</td><td>All</td></tr>
<tr><td>1b</td><td>Enable</td><td>Enable</td><td>All</td></tr>
</table></td></tr>
<tr><td>30</td><td><b>Pipe_State</b>
Project: All
Default Value: 0b
This <b>read only</b> bit indicates the actual state of the pipe. Since there can be some delay between disabling the pipe and the pipe actually shutting off, this bit indicates the true current state of the pipe.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>Disabled</td><td>Pipe is disabled</td><td>All</td></tr>
<tr><td>1b</td><td>Enabled</td><td>Pipe is enabled</td><td>All</td></tr>
</table></td></tr>
<tr><td>29</td><td><b>Reserved</b> Project: All Format:</td></tr>
<tr><td>26</td><td><b>Reserved</b> Project: All Format:</td></tr>
</table>

PIPE_CONF

25:24

Pipe_Palette_Gamma_Mode

Project: All

Default Value: 0b

These bits select which mode the pipe gamma correction logic works in. See the Display Palette Registers for information on the different palette/gamma modes. Other gamma units such as in the sprite are unaffected by this bit.

Value	Name	Description	Project
00b	8 bit	8-bit Legacy Palette Mode	All
01b	10 bt	10-bit Precision Palette Mode	All
10b	12 bit	12-bit Interpolated Gamma Mode	All
11b	Split	Split Gamma Mode (separate pipe gamma functions before and after pipe CSC)	All

23:21

Interlaced_Mode

Project: All

Default Value: 0b

These bits are used for software control of the pipe interlaced mode. Hardware controlled interlacing can be selected in Hardware_Controlled_Refresh_Rate_Select.

Note: VGA display modes do not work while in interlaced fetch modes

Value	Name	Description	Project
000b	PF-PD	Progressive Fetch with Progressive Display (Hardware controlled refresh rate switching can be enabled)	All
001b	PF-ID	Progressive Fetch with Interlaced Display **Requires 7x5 capable panel fitter to be enabled**	All
011b	IF-ID	Interlaced Fetch with Interlaced Display	All
Others	Reserved	Reserved	All

20

Display_Power_Mode_Switch

Project: All

Default Value: 0b

This bit selects the the software controlled progressive-to-progressive power saving mode (software controlled DRRS). Hardware_Controlled_Refresh_Rate_Select must be disabled when enabling this.

Link and data M/N 1 values are used for normal settings, M/N 2 values for low power settings. Pixel clock FP0 values are used for normal settings, FP1 values for low power settings.

Value	Name	Description	Project
0b	Normal	Normal progressive refresh rate	All
1b	Low Power	Low power progressive refresh rate	All

PIPE_CONF

19:18 **MSA_Timing_Delay**

Project: All

Default Value: 00b

Value	Name	Description	Project
00b	Line1	MSA and sDRRS timing switch occur within the first line of vertical blank	All
01b	Line2	MSA and sDRRS timing switch occur within the second line of vertical blank	All
10b	Line3	MSA and sDRRS timing switch occur within the third line of vertical blank	All
11b	Line4	MSA and sDRRS timing switch occur within the fourth line of vertical blank	All

17:16 **Reserved** Project: All Format: MBZ

15:14 **Display_Rotation_Info**

Project: All

Default Value: 0b

These are informative bits set by software to indicate this pipe is being rotated. Software should set these for both hardware and software rotation cases. Hardware rotation is **not** enabled through these bits.

Value	Name	Description	Project
00b	None	No rotation on this pipe	All
01b	90	90° rotation on this pipe	All
10b	180	180° rotation on this pipe	All
11b	270	270° rotation on this pipe	All

13 **Color_Range_Select**

Project: All

Default Value: 0b

This bit is used to select the color range of outputs. When CE color range is selected the pipe output will be compressed and offset to the CE range.

Value	Name	Description	Project
0b	Full	Apply full color range to the output	All
1b	CE	Apply CE color range to the output	All

<table>
<tr><th colspan="2">PIPE_CONF</th></tr>
<tr><td>12:11</td><td>
<b>Pipe_output_color_space_select</b>
Project: All
Default Value: 0b
Informs the ports of the pipe output color space. Plane data formats and CSC need to be programmed to match what is selected here.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>00b</td><td>RGB</td><td>RGB</td><td>All</td></tr>
<tr><td>01b</td><td>YUV 601</td><td>YUV 601</td><td>All</td></tr>
<tr><td>10b</td><td>YUV 709</td><td>YUV 709</td><td>All</td></tr>
<tr><td>11b</td><td>Reserved</td><td>Reserved</td><td>All</td></tr>
</table>
</td></tr>
<tr><td>10</td><td>
<b>xcYCC_Color_Range_Limit</b>
Project: All
Default Value: 0b
This bit is used to limit the color range of the port outputs from 1 to 254 for 8-bit components, 4 to 109 for 10bit components, or 16 to 4079 for 12-bit components. Values outside of the range will be clamped to fit within the range. There is no need to set the equivalent bit in the south display transcoder configuration register if the bit is set in this register.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>Full</td><td>Do not limit the range</td><td>All</td></tr>
<tr><td>1b</td><td>Limit</td><td>Limit range</td><td>All</td></tr>
</table>
</td></tr>
<tr><td>9</td><td><b>Reserved</b> Project: All Format: MBZ</td></tr>
<tr><td>8</td><td>
<b>BFI_enable</b>
Project: All
Default Value: 0b
This bit enables black frame insertion on this pipe. This bit should not be changed while the pipe or port are enabled.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>Disable</td><td>BFI disabled</td><td>All</td></tr>
<tr><td>1b</td><td>Enable</td><td>BFI enabled</td><td>All</td></tr>
</table>
</td></tr>
</table>

PIPE_CONF

7:5

Bits_Per_Color

Project: All

Default Value: 0b

This field selects the number of bits per color output on ports connected to this pipe. Software should enable dithering if selecting a pixel color depth higher or lower than the pixel color depth of the frame buffer.

Value	Name	Description	Project
000b	8 bpc	8 bits per color	All
001b	10 bpc	10 bits per color	All
010b	6 bpc	6 bits per color	All
011b	12 bpc	12 bits per color	All
Others	Reserved	Reserved	All

4

Dithering_enable

Project: All

Default Value: 0b

This bit enables dithering

Value	Name	Description	Project
0b	Disable	Dithering disabled	All
1b	Enable	Dithering enabled	All

3:2

Dithering_type

Project: All

Default Value: 0b

These bits select dithering type.

Value	Name	Description	Project
00b	Spatial	Spatial	All
01b	ST1	Spatio-Temporal 1	All
10b	ST2	Spatio-Temporal 2 (test mode)	All
11b	Temporal	Temporal only (test mode)	All

1:0

Reserved Project: All Format: MBZ

4.1.3 PIPE_CLFR_CTL—Pipe CLFR Control

PIPE_CLFR_CTL	
Register Type:	MMIO
Project:	All
Default Value:	000000FFh
Access:	R/W
Size (in bits):	32

Bit	Description
31	**Pipe_Content_Locked_FR_Enable** Project: All Default Value: 0b Setting this bit to the value of one enables content locked frame rate on this pipe.
30:29	**Flip_Source** Project: All Default Value: 00b These bits set the flip command to which the frame rate is locked.

Bit 31 values:

Value	Name	Description	Project
0b	Disable	Disable	All
1b	Enable	Enable	All

Bits 30:29 values:

Value	Name	Description	Project
00b	Disable	Disable State	All
01b	Primary	Lock to primary on this pipe	All
10b	Sprite	Lock to sprite on this pipe	
11b	Reserved	Reserved	

PIPE_CLFR_CTL

28:26 **Threshold**

Project: All

Default Value: 000b

The difference in flip line number between consecutive flip requests must be less than the threshold value to be counted in the frame rate calculation.

Value	Name	Description	Project
000b	16	16 lines	All
001b	32	32 lines	All
010b	48	48 lines	All
011b	64	64 lines	All
100	80	80 lines	
101	96	96 lines	
110	112	112 lines	
111	Disable	Disable threshold checking	

25:23 **Average_Weighting_W1** Project: All Format:

Current sample weighting factor to be included in the running average for flip line calculation.

22:20 **Adjustment_Weighting_W2** Project: All Format:

The tracking velocity to be used in correction of Vtotal, in percentage.

19:14 **Vtotal_Tolerance_High_Limit** Project: All Format:

This setting indicate the number of blank lines that may be added to the display frame to achieve content locked frame rate. Software must program these bits to stay within the acceptable tolerance of the display rate desired.

Equation:

High Vtotal limit = pixel clk high limit * Vtotal / pixel clk actual

Vtotal tolerance high limit = High Vtotal limit - Vtotal (programmed)

13:8 **Vtotal_Tolerance_Low_Limit** Project: All Format:

This setting indicate the number of blank lines that may be subtracted from the display frame to achieve content locked frame rate. Software must program these bits to stay within the acceptable tolerance of the display rate desired.

Equation:

Low Vtotal limit = pixel clk low limit * Vtotal / pixel clk actual

Vtotal tolerance low limit = Vtotal (programmed) - Low Vtotal limit

7:0 **Frame_Out_of_Range_Count** Project: All Format:

These bits indicate the number of frames with a flip request out of range tolerated before the averaging function is reset. Please note that the counter is reset each time a frame counter value is sent to the averaging function. When programmed to 0xFF the averaging reset is disabled

4.1.4 PIPE_FRMCNT—Pipe Frame Count

PIPE_FRMCNT	
Register Type:	MMIO
Project:	All
Default Value:	00000000h
Access:	Read Only
Size (in bits):	32

Bit	Description
31:0	**Pipe_Frame_Counter** Project: All Format: Provides read back of the display pipe frame counter. This counter increments on every start of vertical blank and rolls over back to 0 after (2^32)-1 frames.

4.1.5 PIPE_FLIPCNT—Pipe Flip Count

PIPE_FLIPCNT	
Register Type:	MMIO
Project:	All
Default Value:	00000000h
Access:	Read Only
Size (in bits):	32
Bit	**Description**
31:0	**Pipe_Flip_Counter** Project: All Format: Provides read back of the display pipe flip counter. This counter increments on each flip of the surface of the primary plane on this pipe. This includes command streamer asynchronous and synchronous flips and MMIO writes to the primary plane surface address. It rolls over back to 0 after (2^32)-1 flips.

4.1.6 PIPE_FRMTMSTMP—Pipe Frame Time Stamp

PIPE_FRMTMSTMP	
Register Type:	MMIO
Project:	All
Default Value:	00000000h
Access:	R/W
Size (in bits):	32
Bit	**Description**
31:0	**Pipe_Frame_Time_Stamp** Project: All Format: Provides read back of the display pipe frame time stamp. The time stamp value is sampled at every start of vertical blank. The TIMESTAMP_CTR register has information on the time stamp value.

4.1.7 PIPE_FLIPTMSTMP—Pipe Flip Time Stamp

PIPE_FLIPTMSTMP	
Register Type:	MMIO
Project:	All
Default Value:	00000000h
Access:	R/W
Size (in bits):	32

Bit	Description
31:0	**Pipe_Flip_Time_Stamp** Project: All Format: Provides read back of the display pipe flip time stamp. The time stamp value is sampled on each flip of the surface of the primary plane on this pipe. This includes command streamer asynchronous and synchronous flips and MMIO writes to the primary plane surface address. The TIMESTAMP_CTR register has information on the time stamp value.

4.2 Cursor Plane

The CUR_CTL and CUR_FBC_CTL active registers will be updated on the vertical blank or when pipe is disabled, after the CUR_BASE or CUR_POPUPBASE register is written, or when cursor is not yet enabled – thus providing an atomic update of those registers together with the CUR_BASE or CUR_POPUPBASE register.

4.2.1 CUR_CTL—Cursor Control

CUR_CTL	
Register Type:	MMIO
Project:	All
Default Value:	00000000h
Access:	R/W
Size (in bits):	32
Double Buffer Update Point:	Start of vertical blank or pipe disabled or cursor disabled, after armed
Double Buffer Armed By:	Write to CUR_BASE or CUR_POPUPBASE
The cursor is enabled by programming a valid cursor mode in the cursor mode select fields. The cursor is disabled by programming all 0s in the cursor mode select fields.	

<table>
<tr><th colspan="2">CUR_CTL</th></tr>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31:28</td><td><b>Reserved</b> Project: All Format:</td></tr>
<tr><td>27</td><td><b>Popup_Cursor_Enabled</b>
Project: All
Default Value: 0b
Popup cursor may only be enabled on a pipe on which VGA is enabled. When in popup mode, hardware interprets the cursor base address as a physical address instead of a graphics address. Only 2bpp cursor data formats are allowed with VGA popup.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>Hi-Res</td><td>Cursor is hi-res</td><td>All</td></tr>
<tr><td>1b</td><td>VGA</td><td>Cursor is VGA popup</td><td>All</td></tr>
</table></td></tr>
<tr><td>26</td><td><b>Gamma_Enable</b>
Project: All
Default Value: 0b
This bit enables pipe gamma correction for the cursor pixel data. In VGA pop-up operation, the cursor data will always bypass gamma.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>Disable</td><td>Cursor pixel data bypasses pipe gamma correction</td><td>All</td></tr>
<tr><td>1b</td><td>Enable</td><td>Cursor pixel data passes through pipe gamma correction</td><td>All</td></tr>
</table></td></tr>
<tr><td>25</td><td><b>Reserved</b> Project: All Format:</td></tr>
<tr><td>24</td><td><b>Pipe_CSC_Enable</b>
Project: All
Default Value: 0b
This bit enables pipe color space conversion for the cursor pixel data.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>Disable</td><td>Cursor pixel data bypasses pipe color space conversion</td><td>All</td></tr>
<tr><td>1b</td><td>Enable</td><td>Cursor pixel data passes through pipe color space conversion</td><td>All</td></tr>
</table></td></tr>
<tr><td>23:16</td><td><b>Reserved</b> Project: All Format:</td></tr>
<tr><td>15</td><td><b>180_Rotation</b>
Project: All
Default Value: 0b
This mode causes the cursor image to be rotated 180°. In addition to setting this bit, software must also adjust the cursor position to match the physical orientation of the display.
Only 32 bits per pixel cursors can be rotated. This field must be zero when the cursor format is 2 bits per pixel.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>None</td><td>No rotation</td><td>All</td></tr>
<tr><td>1b</td><td>180</td><td>180° rotation (only for 32 bit per pixel cursors)</td><td>All</td></tr>
</table></td></tr>
</table>

CUR_CTL

14 **Trickle_Feed_Enable**

Project: All

Default Value: 0b

Value	Name	Description	Project
0b	Enable	Trickle Feed Enabled - Data requests are sent whenever there is space in the Display Data Buffer	All
1b	Disable	Trickle Feed Disabled - Data requests are sent in bursts	All

13:12 **Reserved** Project: All Format:

11:10 **Force_Alpha_Plane_Select**

Project: All

Default Value: 00b

This field selects which planes the cursor alpha value will be forced for. It is used together the the Force_Alpha_Value field.

Value	Name	Description	Project
00b	Disable	Disable alpha forcing	All
01b	Sprite	Enable alpha forcing where cursor overlaps sprite pixels	All
10b	Primary	Enable alpha forcing where cursor overlaps primary pixels	All
11b	Both	Enable alpha forcing where cursor overlaps either sprite or primary pixels.	All

9:8 **Force_Alpha_Value**

Project: All

Default Value: 00b

This field controls the behavior of cursor when alpha blending onto certain plane pixels. It does nothing when cursor is not using an alpha source format. It is used together with the Force_Alpha_Plane_Select field.

Value	Name	Description	Project
00b	Disable	Cursor pixels alpha blend normally over any plane	All
01b	50	Cursor pixels with alpha >= 50% are made fully opaque where they overlap the selected plane(s). Cursor pixels with alpha < 50% are made fully transparent where they overlap the selected plane(s).	All
10b	75	Cursor pixels with alpha >= 75% are made fully opaque where they overlap the selected plane(s). Cursor pixels with alpha < 75% are made fully transparent where they overlap the selected plane(s).	All
11b	100	Cursor pixels with alpha = 100% are made fully opaque where they overlap the selected plane(s). Cursor pixels with alpha < 100% are made fully transparent where they overlap the selected plane(s).	All

7:6 **Reserved** Project: All Format:

CUR_CTL

5

Cursor_Mode_Select_5

Project: All

Default Value: 0b

This bit together with bits 2:0 select the mode for cursor as shown in the cursor mode select table below.

Bit 5	Bits 2:0	Mode
0	000	**Cursor is disabled**
0	001	Reserved
0	010	128 x 128 32bpp AND/INVERT See description off 64 x 64 32bpp AND/INVERT format
0	011	256 x 256 32bpp AND/INVERT See description off 64 x 64 32bpp AND/INVERT format
0	100	64 x 64 2bpp Indexed 3-color and transparency mode
0	101	64 x 64 2bpp Indexed AND/XOR 2-plane mode
0	110	64 x 64 2bpp Indexed 4-color mode
0	111	64 x 64 32bpp AND/INVERT Least significant three bytes provides cursor RGB 888 color information Most Significant Byte: All Ones: Opaque, show the cursor color All Zeros: Transparent (color must also equal zero) Other: Invert the underlying display pixel data (ignore the color)
1	000	Reserved
1	001	Reserved
1	010	128 x 128 32bpp ARGB (8:8:8:8 MSB-A:R:G:B)
1	011	256 x 256 32bpp ARGB (8:8:8:8 MSB-A:R:G:B)
1	100	64 x 64 32bpp AND/XOR Least significant three bytes provides cursor RGB 888 color information Most Significant Byte: All Ones: Opaque, show the cursor color All Zeros: Transparent (color must also equal zero) Other: XOR the cursor color with the underlying display pixel data
1	101	128 x 128 32bpp AND/XOR See description off 64 x 64 32bpp AND/XOR format
1	110	256 x 256 32bpp AND/XOR See description off 64 x 64 32bpp AND/XOR format
1	111	64 x 64 32bpp ARGB (8:8:8:8 MSB-A:R:G:B)

Note: 32bpp formats are not allowed with VGA popup

Note: The cursor vertical size can be overriden by the size reduction mode

Note: INVERT, XOR, and alpha blends may not look as expected when the plane underlying the cursor is YUV or extended range RGB. Out of range RGB values will be clamped prior to alpha blending, INVERT, or XOR with cursor. It is recommended to use Force_Alpha when cursor is alpha blending onto an plane of a different color space or extended gamut

4:3 **Reserved** Project: All Format:

2:0

Cursor_Mode_Select_2_0

Project: All

Default Value: 000b

4.2.2 CUR_BASE—Cursor Base Address

CUR_BASE	
Register Type:	MMIO
Project:	All
Default Value:	00000000h
Access:	R/W
Size (in bits):	32
Double Buffer Update Point:	Start of vertical blank or pipe disabled

Writes to this register arm cursor registers for this pipe

This register is only used when cursor is in the hi-res mode. In VGA popup mode CUR_POPUPBASE is used instead and this register must not be written. This register specifies the graphics memory address at which the cursor image data is located.

Bit	Description
31:12	**Cursor_Base_31_12** Project: All Address: GraphicsAddress[31:12] This field specifies bits 31:12 of the graphics address of the base of the cursor for hi-res mode. The cursor surface address must be 4K byte aligned. The cursor must be in linear memory, it cannot be tiled. When performing 180° rotation, this address does not need to change, hardware will internally offset to start from the last pixel of the last line of the cursor.
11:3	**Reserved** Project: All Format:
2	**Decryption_Request** Project: All Default Value: 0b This bit requests decryption to be enabled for this plane. This request will be qualified with the separate decryption allow message in order to create the decryption enable.
1:0	**Reserved** Project: All Format:

Decryption_Request values:

Value	Name	Description	Project
0b	Not requested	Decryption not requested	All
1b	Requested	Decryption requested	All

4.2.3 CUR_POS—Cursor Position

CUR_POS	
Register Type:	MMIO
Project:	All
Default Value:	00000000h
Access:	R/W
Size (in bits):	32
Double Buffer Update Point:	Start of vertical blank or pipe disabled

This register specifies the screen position of the cursor.

For high resolution display modes, the cursor must have at least a single pixel positioned over the pipe source area. For VGA Popup, the cursor must be completely contained within the VGA source area, which can include the VGA border.

The origin of the cursor position is always the upper left corner of the display pipe source image area. When performing 180° rotation, the cursor image is rotated by hardware, but the position is not, so it must be adjusted by software if it is desired to maintain the same apparent position on a physically rotated display.

Bit	Description
31	**Y_Position_Sign** Project: All This specifies the sign of the vertical position of the cursor upper left corner.
30:28	**Reserved** Project: All Format: MBZ
27:16	**Y_Position_Magnitude** Project: All This specifies the magnitude of the vertical position of the cursor upper left corner in lines.
15	**X_Position_Sign** Project: All This specifies the sign of the horizontal position of the cursor upper left corner.
14:12	**Reserved** Project: All Format: MBZ
11:0	**X_Position_Magnitude** Project: All This specifies the magnitude of the horizontal position of the cursor upper left corner in pixels.

4.2.4 CUR_POPUPBASE—Cursor Popup Base Address

CUR_POPUPBASE	
Register Type:	MMIO
Project:	All
Default Value:	00000000h
Access:	R/W
Size (in bits):	32
Double Buffer Update Point:	Start of vertical blank or pipe disabled

Writes to this register arm cursor registers for this pipe
This register is only used when cursor is in the VGA popup mode. In hi-res mode CUR_BASE is used instead and this register must not be written. This register specifies the physical memory address at which the cursor image data is located.

Bit	Description
31:12	**Cursor_Popup_Base_31_12** Project: All Address: PhysicalAddress[31:12] This field specifies bits 31:12 of the physical address of the base of the cursor for VGA popup mode. The cursor surface address must be 4K byte aligned. The cursor must be in linear memory, it cannot be tiled.
11:7	**Reserved** Project: All Format:
6:0	**Cursor_Popup_Base_38_32** Project: All Address: PhysicalAddress[38:32] This field specifies bits 38:32 of the physical address of the base of the cursor for VGA popup mode. See restrictions in Cursor VGA Popup Base Address field.

4.2.5 CUR_PAL—Cursor Palette

Cursor Palette Format	
Project:	All

Bit	Description
31:24	**Reserved** Project: All Format: MBZ

Cursor Palette Format

23:16	**Palette_Red**	Project:	All	Format:		
	These registers specify the cursor palette. The data can be pre-gamma corrected and bypass the pipe gamma correction logic or pass through the pipe gamma correction.					
15:8	**Palette_Green**	Project:	All	Format:		
7:0	**Palette_Blue**	Project:	All	Format:		

CUR_PAL

Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W**
Size (in bits):	**4x32**
Double Buffer Update Point:	**Start of vertical blank or pipe disabled**

The cursor palette provides color information when using the indexed modes. The two-bit index selects one of the four colors or two of the colors when in the AND/XOR cursor mode.

The table below describes the palette usage for different cursor modes and indexes.

Index Value	2 color mode	3 color mode	4 color mode
00	Palette 0	Palette 0	Palette 0
01	Palette 1	Palette 1	Palette 1
10	Transparent	Transparent	Palette 2
11	Invert destination	Palette 3	Palette 3

DWord	Bit	Description					
0	31:0	**CUR_PAL0**	Project:	All	Format:	Cursor Palette Format
1	31:0	**CUR_PAL1**	Project:	All	Format:	Cursor Palette Format
2	31:0	**CUR_PAL2**	Project:	All	Format:	Cursor Palette Format
3	31:0	**CUR_PAL3**	Project:	All	Format:	Cursor Palette Format

4.2.6 CUR_FBC_CTL—Cursor FBC Control

CUR_FBC_CTL	
Register Type:	MMIO
Project:	All
Default Value:	00000000h
Access:	R/W
Size (in bits):	32
Double Buffer Update Point:	Start of vertical blank or pipe disabled or cursor disabled, after armed
Double Buffer Armed By:	Write to CUR_BASE or CUR_POPUPBASE

Bit	Description
31	**Size_Reduction_Enable** Project: All Default Value: 0b This enables cursor size reduction logic. The cursor engine will fetch and display the programmed reduced number of lines, then go transparent for the rest of the frame. The reduced scan lines value must be programmed when cursor size reduction is enabled. Cursor size reduction is not allowed with VGA popup, 2bpp cursor formats, or cursor 180° rotation.
30:8	**Reserved** Project: All Format:
7:0	**Reduced_Scan_Lines** Project: All Default Value: 00h This specifies the number of scan lines of cursor data to fetch and display when cursor size reduction is enabled. The value programmed is the size minus one. The minimum size is 8 lines, programmed as 07h. The maximum size can not be greater than the normal size when size reduction is not enabled.

Bit 31 values:

Value	Name	Description	Project
0b	Disable	Disable cursor size reduction	All
1b	Enable	Enable cursor size reduction	All

4.2.7 PLANE_SURFLIVE—Plane Live Base Address

PLANE_SURFLIVE	
Register Type:	MMIO
Project:	All
Default Value:	00000000h
Access:	Read Only
Size (in bits):	32

Bit	Description
31:0	**Live_Surface_Base_Address** Project: All Format: This gives the live value of the surface base address as being currently used for the plane.

4.3 Primary Plane

Many of the plane control active registers will be updated on the vertical blank or when pipe is disabled, after the surface base address register is written, or when the plane is not yet enabled – thus providing an atomic update of those registers together with the surface base address register.

4.3.1 PRI_CTL—Primary Control

<table>
<tr><th colspan="2">PRI_CTL</th></tr>
<tr><td>Register Type:</td><td>MMIO</td></tr>
<tr><td>Project:</td><td>All</td></tr>
<tr><td>Default Value:</td><td>00000000h</td></tr>
<tr><td>Access:</td><td>R/W</td></tr>
<tr><td>Size (in bits):</td><td>32</td></tr>
<tr><td>Double Buffer Update Point:</td><td>Start of vertical blank or pipe disabled or primary disabled, after armed</td></tr>
<tr><td>Double Buffer Armed By:</td><td>Write to PRI_SURF</td></tr>
<tr><td colspan="2"></td></tr>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31</td><td><b>Primary_Plane_Enable</b> Project: All Format: Enable
When this bit is set, the primary plane will generate pixels for display. When set to zero, primary plane memory fetches cease and plane output is transparent. When in Self Refresh Big FIFO mode, a write to enable the plane will be internally buffered and delayed while Big FIFO mode is exiting.</td></tr>
<tr><td>30</td><td><b>Gamma_Enable</b>
Project: All
Default Value: 0b
This bit enables pipe gamma correction for the plane pixel data. For 8-bit indexed display data, this bit should be set to a one.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>Disable</td><td>Plane pixel data bypasses pipe gamma correction</td><td>All</td></tr>
<tr><td>1b</td><td>Enable</td><td>Plane pixel data passes through pipe gamma correction</td><td>All</td></tr>
</table></td></tr>
</table>

PRI_CTL

29:26 **Source_Pixel_Format**

Project: All

Default Value: 0b

This field selects the source pixel format for the primary plane.

The 8-bpp indexed format will use the pipe palette.

Before entering the blender, each source format is converted to the pipe pixel format.

Alpha values are ignored.

Value	Name	Description	Project
0010b	8-bit Indexed	8-bit Indexed	All
0101b	16-bit BGRX 5:6:5	16-bit BGRX (5:6:5 MSB-R:G:B)	All
0110b	32-bit BGRX 8:8:8	32-bit BGRX (8:8:8:8 MSB-X:R:G:B)	All
1000b	32-bit RGBX 10:10:10	32-bit RGBX (2:10:10:10 MSB-X:B:G:R)	All
1001b	32-bit XR_BIAS RGBX 10:10:10	32-bit Extended Range Bias RGBX (2:10:10:10 MSB-X:B:G:R)	All
1010b	32-bit BGRX 10:10:10	32-bit BGRX (2:10:10:10 MSB-X:R:G:B)	All
1100b	64-bit RGBX FP	64-bit RGBX Floating Point(16:16:16:16 MSB-X:B:G:R)	All
1110b	32-bit RGBX 8:8:8	32-bit RGBX (8:8:8:8 MSB-X:B:G:R)	All
Others	Reserved	Reserved	All

25 **Reserved** Project: All Format:

24 **Pipe_CSC_Enable**

Project: All

Default Value: 0b

This bit enables pipe color space conversion for the plane pixel data.

Value	Name	Description	Project
0b	Disable	Plane pixel data bypasses the pipe color space conversion	All
1b	Enable	Plane pixel data passes through the pipe color space conversion	All

23:16 **Reserved** Project: All Format:

15 **180_Display_Rotation**

Project: All

Default Value: 0b

This mode causes the plane image to be rotated 180°. In addition to setting this bit, software must also set the surface address offset (lineary or tiled offset registers depending on tiled surface select) to the lower right corner of the unrotated image.

Value	Name	Description	Project
0b	None	No rotation	All
1b	180	180° rotation	All

PRI_CTL

14 **Trickle_Feed_Enable**

Project: All

Default Value: 0b

Value	Name	Description	Project
0b	Enable	Trickle Feed Enabled - Plane data requests are sent whenever there is space in the Display Data Buffer.	All
1b	Disable	Trickle Feed Disabled - Plane data requests are sent in bursts	All

13:11 **Reserved** Project: All Format:

10 **Tiled_Surface**

Project: All

Default Value: 0b

This bit indicates that the surface data is in tiled memory. The tile pitch is specified in bytes in the plane stride register. Only X tiling is supported.

When this bit is set, it affects the interpretation of the offset and surface address registers.

This bit may be updated through MMIO writes or through a command streamer initiated synchronous flip.

Value	Name	Description	Project
0b	Linear	Plane uses linear memory	All
1b	X-Tiled	Planes uses X-Tiled memory	All

9 **Async_Address_Update_Enable**

Project: All

Default Value: 0b

This bit will enable asynchronous updates of the plane surface address when written by MMIO (MMIO asynchronous flips). The surface address will change with the next plane TLB request or when start of vertical blank is reached. Updates during vertical blank may not complete until after the first few active lines are displayed.

Restrictions:

- No command streamer initiated flips to this plane are allowed when this bit is enabled.
- Wait for flip done indication before writing the surface address register again.
- Only the plane surface address register can be changed asynchronously

Value	Name	Description	Project
0b	Sync	Surface Address MMIO writes will update synchronous to start of vertical blank (synchronous flips)	All
1b	Async	Surface Address MMIO writes will update asynchronously (asynchronous flips)	All

8:0 **Reserved** Project: All Format: MBZ

Plane Source Pixel Format Mapping of Bits to Colors:

Format	Ignored	Red	Green	Blue
16-bit BGRX 5:6:5	N/A	15:11	10:5	4:0
32-bit BGRX 8:8:8	31:24	23:16	15:8	7:0
32-bit RGBX 10:10:10	31:30	9:0	19:10	29:20
32-bit BGRX 10:10:10	31:30	29:20	19:10	9:0
64-bit RGBX Float 16:16:16 Each component is 1:5:10 MSb-sign:exponent:fraction	63:48	15:0	31:16	47:32
32-bit RGBX 8:8:8	31:24	7:0	15:8	23:16
32-bit XR_BIAS RGBX 10:10:10	31:30	9:0	19:10	29:20

4.3.2 PRI_LINOFF—Primary Linear Offset

PRI_LINOFF	
Register Type:	MMIO
Project:	All
Default Value:	00000000h
Access:	R/W
Size (in bits):	32
Double Buffer Update Point:	Start of vertical blank or pipe disabled

Bit	Description
31:0	**Linear_Offset** Project: All Format: This register specifies the panning for the plane surface in linear memory. This value is added to the surface address to get the address of the first pixel to be displayed. This offset must be at least pixel aligned for RGB formats. When performing 180° rotation, the unpanned offset must be the difference between the last pixel of the last line of the display data in its unrotated orientation and the display surface address. When the surface is tiled, the tiled offset is programmed and the contents of this register are ignored.

4.3.3 PRI_STRIDE—Primary Stride

PRI_STRIDE	
Register Type:	MMIO
Project:	All
Default Value:	00000000h
Access:	R/W
Size (in bits):	32
Double Buffer Update Point:	Start of vertical blank or pipe disabled or primary disabled, after armed
Double Buffer Armed By:	Write to PRI_SURF

Bit	Description
31:16	**Reserved** Project: All Format:
15:6	**Stride** Project: All Format: This is the stride for the plane in bytes. This value is used to determine the line to line increment for the plane. When using linear memory, this must be at least 64 byte aligned. When using tiled memory, this must be at least 512 byte aligned. This register may be updated through MMIO writes or through a command streamer initiated synchronous flip. The stride is limited to a maximum of 32K bytes.
5:0	**Reserved** Project: All Format:

4.3.4 PRI_SURF—Primary Surface Base Address

PRI_SURF	
Register Type:	MMIO
Project:	All
Default Value:	00000000h
Access:	R/W
Size (in bits):	32
Double Buffer Update Point:	Start of vertical blank, pipe disabled, or next plane line request if asynchronous flip
Writes to this register arm primary registers for this pipe	

PRI_SURF	
Bit	**Description**
31:12	**Surface_Base_Address** Project: All Address: GraphicsAddress[31:12] This address specifies the surface base address. It represents an offset from the graphics memory aperture base and is mapped to physical pages through the global GTT. It must be at least 4KB aligned. When performing asynchronous flips and the display surface is in tiled memory, this address must be 256KB aligned. This register may be updated through MMIO writes or through a command streamer initiated synchronous or asynchronous flip.
11:3	**Reserved** Project: All Format:
2	**Decryption_Request** Project: All Default Value: 0b This bit requests decryption to be enabled for this plane. This request will be qualified with the separate decryption allow message in order to create the decryption enable. This bit is only allowed to change on a synchronous update, but once set with a synchronous update, the bit can remain set while using asynchronous updates.
1:0	**Reserved** Project: All Format:

Bit 2 values:

Value	Name	Description	Project
0b	Not requested	Decryption not requested	All
1b	Requested	Decryption requested	All

4.3.5 PRI_TILEOFF—Primary Tiled Offset

PRI_TILEOFF	
Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W**
Size (in bits):	**32**
Double Buffer Update Point:	**Start of vertical blank or pipe disabled**

This register specifies the panning for the plane surface in tiled memory. When the surface is in linear memory, the linear offset is programmed and the contents of this register are ignored. When the surface is tiled, the start position is specified in this register as a (x, y) offset from the beginning of the surface. When performing 180° rotation, the unpanned offset must be programmed to the last pixel of the last line of the display data.

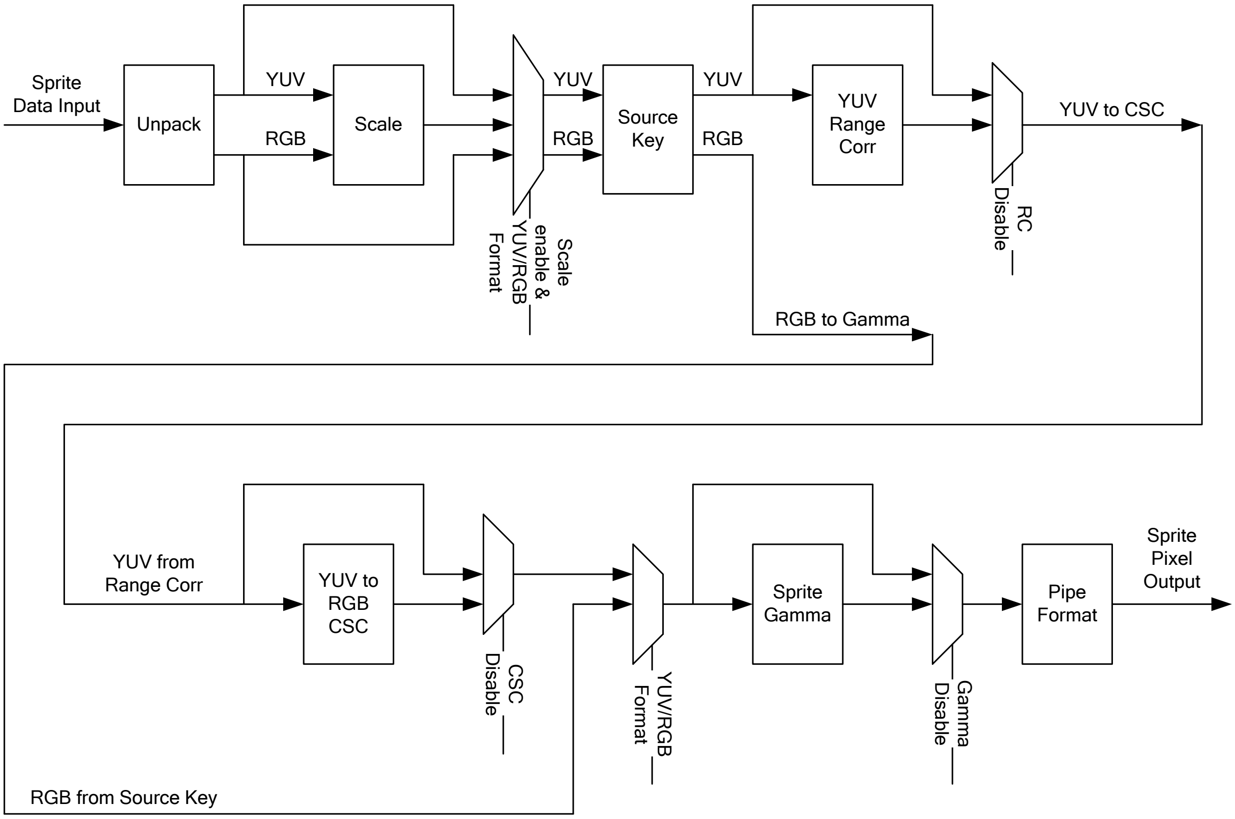
PRI_TILEOFF	
Bit	**Description**
31:28	**Reserved** Project: All Format: MBZ
27:16	**Start_Y_Position** Project: All Format: The vertical offset in lines of the beginning of the active display plane relative to the display surface.
15:12	**Reserved** Project: All Format: MBZ
11:0	**Start_X_Position** Project: All Format: The horizontal offset in pixels of the beginning of the active display plane relative to the display surface.

4.4 Sprite Plane

Many of the plane control active registers will be updated on the vertical blank or when pipe is disabled, after the surface base address register is written, or when the plane is not yet enabled – thus providing an atomic update of those registers together with the surface base address register.

Data flow through the sprite plane (Steps 2-6 may be enabled or disabled by programming control bits):

1. Unpack data into pixels
2. Scale
3. Source Key
4. YUV Range Correction (can only be used by YUV source pixel formats)
5. YUV to RGB Color Space Conversion (can only be used by YUV source pixel formats)
6. Sprite Gamma Correction
7. Conversion to pipe data format


Sprite
Data Input
Unpack
YUV
RGB
Scale
YUV
RGB
Scale
enable &
YUV/RGB
Format
Source
Key
YUV
RGB
YUV
Range
Corr
RC
Disable
YUV to CSC
RGB to Gamma
YUV from
Range Corr
YUV to
RGB
CSC
CSC
Disable
YUV/RGB
Format
Sprite
Gamma
Gamma
Disable
Pipe
Format
Sprite
Pixel
Output
RGB from Source Key

4.4.1 SPR_CTL—Sprite Control

<table>
<tr><th colspan="2">SPR_CTL</th></tr>
<tr><td colspan="2">Register Type: MMIO
Project: All
Default Value: 00000000h
Access: R/W
Size (in bits): 32
Double Buffer Update Point: Start of vertical blank or pipe disabled or sprite disabled, after armed
Double Buffer Armed By: Write to SPR_SURF</td></tr>
<tr><td colspan="2"></td></tr>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31</td><td><b>Sprite_Enable</b> Project: All Format: Enable
When this bit is set, the sprite plane will generate pixels for display. When set to zero, sprite plane memory fetches cease and plane output is transparent. When in Self Refresh Big FIFO mode, a write to enable the plane will be internally buffered and delayed while Big FIFO mode is exiting.</td></tr>
<tr><td>30</td><td><b>Pipe_Gamma_Enable</b>
Project: All
Default Value: 0b
This bit enables pipe gamma correction for the sprite pixel data.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>Disable</td><td>Plane pixel data bypasses pipe gamma correction</td><td>All</td></tr>
<tr><td>1b</td><td>Enable</td><td>Plane pixel data passes through pipe gamma correctopm</td><td>All</td></tr>
</table></td></tr>
<tr><td>29</td><td><b>Reserved</b> Project: All Format: MBZ</td></tr>
<tr><td>28</td><td><b>YUV_Range_Correction_Disable</b>
Project: All
Default Value: 0b
Setting this bit disables the YUV range correction logic inside the sprite. The range correction logic is used to expand the compressed range YUV to full range YUV. The Y channel is expanded from the 8 bit +16 to +235 range to full range. The U and V channels are expanded from the 8 bit -112 to +112 range to full range. Extended range values will be preserved after the expansion. This bit has no effect on RGB source pixel formats since they automatically bypass range correction.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>Enable</td><td>Range correction enabled</td><td>All</td></tr>
<tr><td>1b</td><td>Disable</td><td>No range correction</td><td>All</td></tr>
</table></td></tr>
</table>

SPR_CTL

27:25 **Source_Pixel_Format**

Project: All

Default Value: 0b

This field selects the source pixel format for the sprite plane.

Before entering the blender, each source format is converted to the pipe pixel format.

Alpha values are ignored.

YUV 4:2:2 byte order is programmed separately.

YUV 4:4:4 byte order is not programmable.

RGB color order is programmed separately, except RGB XR_BIAS byte order is not programmable.

Value	Name	Description	Project
000b	YUV 16-bit 4:2:2	YUV 16-bit 4:2:2 packed	All
001b	RGB 32-bit 2:10:10:10	RGB 32-bit 2:10:10:10	All
010b	RGB 32-bit 8:8:8:8	RGB 32-bit 8:8:8:8	All
011b	RGB 64-bit 16:16:16:16	RGB 64-bit 16:16:16:16 Floating Point	All
100b	YUV 32-bit 4:4:4	YUV 32-bit 4:4:4 packed (8:8:8:8 MSB-X:Y:U:V)	All
101b	RGB 32-bit XR_BIAS 10:10:10	RGB 32-bit Extended Range Bias RGBX (2:10:10:10 MSB-X:B:G:R)	All
Others	Reserved	Reserved	All

24 **Pipe_CSC_Enable**

Project: All

Default Value: 0b

This bit enables pipe color space conversion for the plane pixel data. This is separate from the color conversion logic within the sprite plane.

Value	Name	Description	Project
0b	Disable	Plane pixel data bypasses the pipe color space conversion	All
1b	Enable	Plane pixel data passes through the pipe color space conversion	All

23 **Reserved** Project: All Format: MBZ

22 **Sprite_Source_Key_Enable**

Project: All

Default Value: 0b

This bit enables source color keying. Sprite pixel values that match (within range) the key will become transparent. Source key can not be enabled if destination key is enabled.

Value	Name	Description	Project
0b	Disable	Sprite source key is disabled	All
1b	Enable	Sprite source key is enabled	All

21 **Reserved** Project: All Format: MBZ

SPR_CTL

20

RGB_Color_Order

Project: All

Default Value: 0b

This field is used to select the color order when using RGB data formats, except RGB 32-bit XR_BIAS 10:10:10. For other formats, this field is ignored.

Value	Name	Description	Project
0b	BGRX	BGRX (MSB-X:R:G:B)	All
1b	RGBX	RGBX (MSB-X:B:G:R)	All

19

Sprite_YUV_to_RGB_CSC_Dis

Project: All

Default Value: 0b

This bit controls the sprite internal YUV to RGB color space conversion. RGB source pixel formats automatically bypass the sprite internal color space conversion.

Value	Name	Description	Project
0b	Enable	YUV pixel data goes through the sprite color conversion	All
1b	Disable	YUV pixel data bypasses the sprite color conversion	All

18

Sprite_YUV_to_RGB_CSC_Format

Project: All

Default Value: 0b

This bit specifies the source YUV format for the sprite internal YUV to RGB color space conversion operation. This field is ignored when source data is RGB.

Value	Name	Description	Project
0b	BT.601	ITU-R Recommendation BT.601	All
1b	BT.709	ITU-R Recommendation BT.709	All

17:16

YUV_422_Byte_Order

Project: All

Default Value: 0b

This field is used to select the byte order when using YUV 4:2:2 data formats. For other formats, this field is ignored.

Value	Name	Description	Project
00b	YUYV	YUYV (8:8:8:8 MSB-V:Y_2:U:Y_1)	All
01b	UYVY	UYVY (8:8:8:8 MSB-Y_2:V:Y_1:U)	All
10b	YVYU	YVYU (8:8:8:8 MSB-U:Y_2:V:Y_1)	All
11b	VYUY	VYUY (8:8:8:8 MSB-Y_2:U:Y_1:V)	All

SPR_CTL

15

180_Display_Rotation

Project: All

Default Value: 0b

This mode causes the plane image to be rotated 180°. In addition to setting this bit, software must also set the surface address offset (linear or tiled offset registers depending on tiled surface select) to the lower right corner of the unrotated surface image and adjust the plane position to match the physical orientation of the display.

Value	Name	Description	Project
0b	None	No rotation	All
1b	180	180° rotation	All

14

Trickle_Feed_Enable

Project: All

Default Value: 0b

Value	Name	Description	Project
0b	Enable	Trickle Feed Enabled - Data requests are sent whenever there is space in the Display Data Buffer	All
1b	Disable	Trickle Feed Disabled - Data requests are sent in bursts.	All

13

Sprite_Gamma_Disable

Project: All

Default Value: 0b

This bit controls sprite internal gamma correction.

Value	Name	Description	Project
1b	Disable	Disable sprite internal gamma correction	All
0b	Enable	Enable sprite internal gamma correction	All

12:11 **Reserved** Project: All Format:

10

Tiled_Surface

Project: All

Default Value: 0b

This bit indicates that the surface data is in tiled memory. The tile pitch is specified in bytes in the plane stride register. Only X tiling is supported.

When this bit is set, it affects the interpretation of the offset and surface address registers.

This bit may be updated through MMIO writes or through a command streamer initiated synchronous flip.

Value	Name	Description	Project
0b	Linear	Plane uses linear memory	All
1b	X-Tiled	Planes uses X-Tiled memory	All

9:3 **Reserved** Project: All Format: MBZ

SPR_CTL	
2	**Sprite_Destination_Key** Project: All Default Value: 0b This bit enables the destination key function. When blending together sprite and primary planes, if the primary plane pixel matches the key value, then the sprite pixel is output, otherwise the primary pixel is output. Destination Key can not be enabled if source key is enabled. **Value** | **Name** | **Description** | **Project** 0b | Disable | Destination Key is disabled | All 1b | Enable | Destination Key is enabled | All
1:0	**Reserved** Project: All Format: MBZ

Sprite Source Pixel Format Mapping of Bits to Colors:
Note: For RGB formats, see the primary plane source pixel format mapping table

SPRITE YUV 4:2:2	**Y1**	**U**	**Y2**	**V**
YUV 4:2:2 YUYV	7:0	15:8	23:16	31:24
YUV 4:2:2 UYVY	15:8	7:0	31:24	23:16
YUV 4:2:2 YVYU	7:0	31:24	23:16	15:8
YUV 4:2:2 VYUY	15:8	23:16	31:24	7:0
SPRITE YUV 4:4:4	**Ignored**	**Y**	**U**	**V**
YUV 32-bit 4:4:4	31:24	23:16	15:8	7:0

4.4.2 SPR_LINOFF—Sprite Linear Offset

SPR_LINOFF	
Register Type:	MMIO
Project:	All
Default Value:	00000000h
Access:	R/W
Size (in bits):	32
Double Buffer Update Point:	Start of vertical blank or pipe disabled
Bit	Description
31:0	**Linear_Offset** Project: All Format: This register specifies the panning for the plane surface in linear memory. This value is added to the surface address to get the address of the first pixel to be displayed. This offset must be at least pixel aligned for RGB or YUV 4:4:4 formats and even pixel aligned for YUV 4:2:2 formats. When performing 180° rotation, the unpanned offset must be the difference between the last pixel of the last line of the display data in its unrotated orientation and the display surface address. When the surface is tiled, the tiled offset is programmed and the contents of this register are ignored.

4.4.3 SPR_STRIDE—Sprite Stride

SPR_STRIDE	
Register Type:	MMIO
Project:	All
Default Value:	00000000h
Access:	R/W
Size (in bits):	32
Double Buffer Update Point:	Start of vertical blank or pipe disabled or sprite disabled, after armed
Double Buffer Armed By:	Write to SPR_SURF
Bit	Description
31:15	**Reserved** Project: All Format:

SPR_STRIDE	
14:6	**Stride** Project: All Format: This is the stride for the plane in bytes. This value is used to determine the line to line increment for the plane. When using linear memory, this must be at least 64 byte aligned. When using tiled memory, this must be at least 512 byte aligned. This register may be updated through MMIO writes or through a command streamer initiated synchronous flip. The stride is limited to a maximum of 16K bytes when sprite scaling is not enabled, 4K bytes when sprite scaling is enabled.
5:0	**Reserved** Project: All Format:

4.4.4 SPR_POS—Sprite Position

SPR_POS	
Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W**
Size (in bits):	**32**
Double Buffer Update Point:	**Start of vertical blank or pipe disabled or sprite disabled, after armed**
Double Buffer Armed By:	**Write to SPR_SURF**

This register specifies the screen position of the sprite.
The sprite must be completely contained within the pipe source area. Pipe source size >= sprite position + sprite size
The origin of the sprite position is always the upper left corner of the display pipe source image area.
When performing 180° rotation, the sprite image is rotated by hardware, but the position is not, so it must be adjusted by software if it is desired to maintain the same apparent position on a physically rotated display

Bit	Description
31:28	**Reserved** Project: All Format: MBZ
27:16	**Y_Position** Project: All Format: This specifies the vertical position of the sprite upper left corner in lines.
15:12	**Reserved** Project: All Format: MBZ
11:0	**X_Position** Project: All Format: This specifies the horizontal position of the sprite upper left corner in pixels.

4.4.5 SPR_SIZE—Sprite Size

SPR_SIZE	
Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W**
Size (in bits):	**32**
Double Buffer Update Point:	**Start of vertical blank or pipe disabled or sprite disabled, after armed**
Double Buffer Armed By:	**Write to SPR_SURF**

This register specifies the size of the sprite. The sprite must be completely contained within the pipe source area. Pipe source size >= sprite position + sprite size
The sprite must be at least one pixel high and one pixel wide.

Bit	Description
31:28	**Reserved** Project: All Format: MBZ
27:16	**Height** Project: All Format: This specifies the height of the sprite in lines. The value in the register is the height minus one.
15:12	**Reserved** Project: All Format: MBZ
11:0	**Width** Project: All Format: This specifies the width of the sprite in pixels. The value in the register is the width minus one. This should be less than or equal to the stride in pixels. The width (prior to minus one) must be even when a YUV 4:2:2 source pixel format is used.

4.4.6 SPR_SURF—Sprite Surface Base Address

SPR_SURF

Register Type:	MMIO
Project:	All
Default Value:	00000000h
Access:	R/W
Size (in bits):	32
Double Buffer Update Point:	Start of vertical blank or pipe disabled

Writes to this register arm sprite registers for this pipe

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31:12</td><td><b>Surface_Base_Address</b>
Project: All
Address: Graphicsdress[31:12]
This address specifies the surface base address. It represents an offset from the graphics memory aperture base and is mapped to physical pages through the global GTT. It must be at least 4KB aligned. This register may be updated through MMIO writes or through a command streamer initiated synchronous flip.</td></tr>
<tr><td>11:3</td><td><b>Reserved</b> Project: All Format:</td></tr>
<tr><td>2</td><td><b>Decryption_Request</b>
Project: All
Default Value: 0b
This bit requests decryption to be enabled for this plane. This request will be qualified with the separate decryption allow message in order to create the decryption enable.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>Not requested</td><td>Decryption not requested</td><td>All</td></tr>
<tr><td>1b</td><td>Requested</td><td>Decryption requested</td><td>All</td></tr>
</table></td></tr>
<tr><td>1:0</td><td><b>Reserved</b> Project: All Format:</td></tr>
</table>

4.4.7 SPR_TILEOFF—Sprite Tiled Offset

SPR_TILEOFF	
Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000000h**
Access:	**R/W**
Size (in bits):	**32**
Double Buffer Update Point:	**Start of vertical blank or pipe disabled**

This register specifies the panning for the plane surface in tiled memory. When the surface is in linear memory, the linear offset is programmed and the contents of this register are ignored. When the surface is tiled, the start position is specified in this register as a (x, y) offset from the beginning of the surface.
When performing 180° rotation, the unpanned offset must be programmed to the last pixel of the last line of the display data.
This offset must be even pixel aligned for YUV 4:2:2 formats.

Bit	Description
31:28	**Reserved** Project: All Format: MBZ
27:16	**Start_Y_Position** Project: All Format: The vertical offset in lines of the beginning of the active display plane relative to the display surface.
15:12	**Reserved** Project: All Format: MBZ
11:0	**Start_X_Position** Project: All Format: The horizontal offset in pixels of the beginning of the active display plane relative to the display surface.

4.4.8 SPR_KEYVAL—Sprite Key Color Value

SPR_KEYVAL	
Register Type:	MMIO
Project:	All
Default Value:	00000000h
Access:	R/W
Size (in bits):	32
Double Buffer Update Point:	Start of vertical blank or pipe disabled

For source key when sprite source is YUV, this register specifies the source key YUV minimum color value to be used together with the YUV maximum color value and the color channel enable bits to determine if the sprite matches the source key color range.

For source key when sprite source is RGB, this register specifies the source key RGB color value to be used together with the color channel enable bits to determine if the sprite matches the source key color.

For destination key, this register specifies the destination key RGB color value to be used together with the RGB mask bits to determine if the primary matches the destination key color.

A key match can only occur for positive pixel values in the 0 to 1 range. Extended range pixel values will not match.

Bit	Description
31:24	**Reserved** Project: All Format: MBZ
23:16	**V_R_Min_Dest_Key_Value** Project: All Format: Specifies the color key minimum value for the sprite V channel source key, the compare value for sprite Red channel source key, or the compare value for the primary Red channel destination key.
15:8	**Y_G_Min_Dest_Key_Value** Project: All Format: Specifies the color key minimum value for the sprite Y channel source key, the compare value for sprite Green channel source key, or the compare value for the primary Green channel destination key.
7:0	**U_B_Min_Dest_Key_Value** Project: All Format: Specifies the color key minimum value for the sprite U channel source key, the compare value for sprite Blue channel source key, or the compare value for the primary Blue channel destination key.

4.4.9 SPR_KEYMSK—Sprite Key Mask

SPR_KEYMSK

Register Type:	MMIO
Project:	All
Default Value:	00000000h
Access:	R/W
Size (in bits):	32
Double Buffer Update Point:	Start of vertical blank or pipe disabled

For source key, this register specifies which channels to perform key color checking on.

For destination key, this register specifies the key mask to be used with the color value bits to determine if the primary plane pixels match the key. A zero bit in the mask indicates that the corresponding bit match failure should be ignored when determining if the pixel matches.

Note that source key and destination key are mutually exclusive modes of operation, they can not be used simultaneously. For the function that is not enabled, the associated bits in this register should be programmed to zeroes.

Bit	Description
31:27	**Reserved** Project: All Format: MBZ
26	**V_R_Source_Key_Channel_Enable** Project: All Format: Enables the V/Red channel for source key color comparison.
25	**Y_G_Source_Key_Channel_Enable** Project: All Format: Enables the Y/Green channel for source key color comparison.
24	**U_B_Source_Key_Channel_Enable** Project: All Format: Enables the U/Blue channel for source key color comparison.
23:16	**R_Dest_Key_Mask_Value** Project: All Format: Specifies the destination color key mask for the Red channel
15:8	**G_Dest_Key_Mask_Value** Project: All Format: Specifies the destination color key mask for the Green channel
7:0	**B_Dest_Key_Mask_Value** Project: All Format: Specifies the destination color key mask for the Blue channel

4.4.10 SPR_KEYMAX—Sprite Key Color Max

SPR_KEYMAX	
Register Type:	MMIO
Project:	All
Default Value:	00000000h
Access:	R/W
Size (in bits):	32
Double Buffer Update Point:	Start of vertical blank or pipe disabled

For source key when sprite source is YUV, this register specifies the source key YUV maximum color value to be used together with the YUV minimum color value and the color channel enable bits to determine if the sprite matches the source key color range.

Bit	Description
31:24	**Reserved** Project: All Format: MBZ
23:16	**V_Source_Key_Max_Value** Project: All Format: Specifies the color key maximum value for the sprite V channel source key
15:8	**Y_Source_Key_Max_Value** Project: All Format: Specifies the color key maximum value for the sprite Y channel source key
7:0	**U_Source_Key_Max_Value** Project: All Format: Specifies the color key maximum value for the sprite U channel source key

4.4.11 SPR_SCALE—Sprite Scaler Control

SPR_SCALE	
Register Type:	MMIO
Project:	All
Default Value:	00000000h
Access:	R/W
Size (in bits):	32
Double Buffer Update Point:	Start of vertical blank or pipe disabled or sprite disabled, after armed
Double Buffer Armed By:	Write to SPR_SURF

SPR_SCALE

This register controls the sprite scaling. When scaling is enabled, the SPR_SIZE register gives the destination (output to pipe) size of the sprite and this register gives the source (input to sprite) size of the sprite, then the source size will be scaled up or down to the destination size.

Sprite scaling should not be enabled with the RGB XR_BIAS 10:10:10 format, RGB 64-bit format, or any YUV format containing extended range data.

Source and destination sizes must be 3x3 (3x6 when interlacing) or greater when scaling is enabled.

Upscaling of any amount is allowed.

Downscaling up to 16X (source/destination) is allowed. Downscaling greater than 2X will involve decimation. Downscaling increases memory bandwidth requirements.

Horizontal downscaling limits the maximum pixel rate allowed as percent of cdclk.

Rules to calculate the allowed pixel rate with scaling:
Start with maximum pixel rate 80% of cdclk.
Subtract 10% per horizontal decimation step (decimation steps at 2x, 4x, 8x, and 16x downscale).
Subtract 10% more if sprite is using the RGB or YUV 4:4:4 data format.
Then divide that by horizontal downscale amount within each decimation step.
The result is the maximum allowed pixel rate as percent of cdclk frequency.

Panel fitting down scaling may further reduce the maximum pixel rate.

Example of pixel rate reduction with scaling:

Scale factor	Decimation amount	YUV 4:2:2 pixel rate %	RGB or YUV 4:4:4 pixel rate %	Comment
1	1	80	70	No scaling
1.5	1	53	46	
1.99	1	40	35	Max downscale before decimation starts
2	2	70	60	
3	2	46	40	
3.99	2	35	30	
4	4	60	50	
6	4	40	33	
7.99	4	30	25	
8	8	50	40	
12	8	33	26	
15.99	8	25	20	Worst case pixel rate
16	16	40	30	Max downscaling allowed

<table>
<tr><th colspan="2">SPR_SCALE</th></tr>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31</td><td><b>Scaling_Enable</b> Project: All Format: Enable
Enables the scaling function. Source width can be no more than 4k bytes. For best picture quality, disable when scaling is not required. When in Self Refresh Big FIFO mode, scaling enable will be masked off while Big FIFO mode is exiting. Scaling should not be left enabled when sprite is disabled.</td></tr>
<tr><td>30:29</td><td><b>Filter_Control</b>
Project: All
Default Value: 0b
Filter selection
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>00b</td><td>Medium</td><td>Medium Filtering</td><td>All</td></tr>
<tr><td>01b</td><td>Enhancing</td><td>Edge Enhancing Filtering</td><td>All</td></tr>
<tr><td>10b</td><td>Softening</td><td>Edge Softening Filtering</td><td>All</td></tr>
<tr><td>11b</td><td>Reserved</td><td>Reserved</td><td>All</td></tr>
</table></td></tr>
<tr><td>28</td><td><b>Field_Offset</b>
Project: All
Default Value: 0b
Select the vertical offset of the filtered data. Software is responsible for updating this to match the surface data.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>0</td><td>Vertical initial phase of 0</td><td>All</td></tr>
<tr><td>1b</td><td>0.5</td><td>Vertical initial phase of 0.5</td><td>All</td></tr>
</table></td></tr>
<tr><td>27</td><td><b>Field_Enable</b>
Project: All
Default Value: 0b
Enable adjustment of the vertical offset of the filtered data.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>Disable</td><td>Off (Vertical initial phase is 1/2 the scale factor)</td><td>All</td></tr>
<tr><td>1b</td><td>Enable</td><td>On (Vertical initial phase is selected by the Field Offset bit)</td><td>All</td></tr>
</table></td></tr>
<tr><td>26:16</td><td><b>Source_Width</b> Project: All Format:
The horizontal size of the source image to be scaled in pixels. Max number of pixels is 2048; minimum is 3. The value programmed is one less than the number of pixels. Source width can be no more than 4k bytes, counting from a 64 byte alignment. The sprite width (actual width, not the width minus one value) is limited to even values when YUV 4:2:2 source pixel format is used.</td></tr>
<tr><td>15:11</td><td><b>Reserved</b> Project: All Format: MBZ</td></tr>
</table>

SPR_SCALE	
10:0	**Source_Height** Project: All Format: The vertical size of the source image to be scaled in lines. If the source is a field, this is the number of lines in the field. Max number of lines is 2048; minimum is 3 (6 when interlacing). The value programmed is one less than the number of lines. The height must be even when sprite scaling is enabled and the pipe has set planes to interlaced fetch. That means the programmed value must be odd.

4.4.12 SPR_GAMC—Sprite Gamma Correction

SPR_GAMC REFERENCE POINT FORMAT

Project:	All

Bit	Description

SPR_GAMC

Register Type:	MMIO
Project:	All
Default Value:	00000000h; 04010040h; 08020080h; 0C0300C0h; 10040100h; 14050140h; 18060180h; 1C0701C0h; 20080200h; 24090240h; 280A0280h; 2C0B02C0h; 300C0300h; 340D0340h; 380E0380h; 3C0F03C0h;
Access:	R/W
Size (in bits):	16x32

These registers are used to determine the characteristics of the gamma correction for the sprite pixel data pre-blending. Additional gamma correction can be done in the display pipe gamma if desired.

The gamma correction curve is represented by specifying a set of gamma entry reference points spaced equally along the curve for values between -1 and 1. For extended values there is an extended gamma entry reference point at the maximum alowed input value.

All input values are clamped to the -3.0 to 3.0 range before the gamma calculation.

For input values greater than or equal to 0 and less than 1.0, the input value is used to linearly interpolate between two adjacent points of the first 17 gamma entries to create the result value. The first 16 entries are stored in SPR_GAMC with 10 bits per color in a 0.10 format with 0 integer and 10 fractional. The 17th entry is stored in the SPR_GAMC16 register with 11 bits per color in a 1.10 format with 1 integer and 10 fractional bits.

For input values greater than or equal to 1.0 and less than or equal to 3.0, the input value is used to linearly interpolate between the 17th and 18th gamma entries to create the result value. The 18th entry is stored in the SPR_GAMC17 register with 12 bits per color in a 2.10 format with 2 integer and 10 fractional bits.

For negative input values, gamma is mirrored along the X-axis, giving the same result as positive input values, except for a negative sign. When gamma input may be negative, the first gamma point should be programmed to a value of 0.0 in order to have a symmetric mirroring.

To program the gamma correction entries, calculate the desired gamma curve for inputs from 0 to 3.0. The curve must be flat or increasing, never decreasing. For inputs of 0 to 1.0, multiply the input value by 16 to find the gamma entry number, then store the desired gamma result in that entry. For inputs greater than 1.0 and less than or equal to 3.0, store the result for an input of 3.0 in the 18th gamma entry (SRP_GAMC17).

The gamma correction registers should only be updated when the sprite is off, otherwise screen artifacts may show temporarily.
Gamma correction can be enabled or disabled through the sprite control register.

See Pipe Gamma for an example gamma curve diagram.

SPR_GAMC REFERENCE POINT FORMAT

DWord	Bit	Description
0	31:0	**GAMC0** Project: All Format: SPR_GAMC REFERENCE POINT FORMAT
1	31:0	**GAMC1** Project: All Format: SPR_GAMC REFERENCE POINT FORMAT
2	31:0	**GAMC2** Project: All Format: SPR_GAMC REFERENCE POINT FORMAT
3	31:0	**GAMC3** Project: All Format: SPR_GAMC REFERENCE POINT FORMAT
4	31:0	**GAMC4** Project: All Format: SPR_GAMC REFERENCE POINT FORMAT
5	31:0	**GAMC5** Project: All Format: SPR_GAMC REFERENCE POINT FORMAT
6	31:0	**GAMC6** Project: All Format: SPR_GAMC REFERENCE POINT FORMAT
7	31:0	**GAMC7** Project: All Format: SPR_GAMC REFERENCE POINT FORMAT
8	31:0	**GAMC8** Project: All Format: SPR_GAMC REFERENCE POINT FORMAT
9	31:0	**GAMC9** Project: All Format: SPR_GAMC REFERENCE POINT FORMAT
10	31:0	**GAMC10** Project: All Format: SPR_GAMC REFERENCE POINT FORMAT
11	31:0	**GAMC11** Project: All Format: SPR_GAMC REFERENCE POINT FORMAT
12	31:0	**GAMC12** Project: All Format: SPR_GAMC REFERENCE POINT FORMAT
13	31:0	**GAMC13** Project: All Format: SPR_GAMC REFERENCE POINT FORMAT
14	31:0	**GAMC14** Project: All Format: SPR_GAMC REFERENCE POINT FORMAT
15	31:0	**GAMC15** Project: All Format: SPR_GAMC REFERENCE POINT FORMAT

4.4.13 SPR_GAMC16—Sprite Gamma Correction Point 16

SPR_GAMC16	
Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000400h; 00000400h; 00000400h;**
Access:	**R/W**
Size (in bits):	**3x32**

These registers are used to determine the 17th reference point (point 16 when counting from 0) for sprite gamma correction. See SPR_GAMC for sprite gamma programming information.

DWord	Bit	Description
0	31:11	**Reserved** Project: All Format: MBZ
	10:0	**GAMC16R** Project: All This value specifies the 17th reference point that is used for the red color channel sprite gamma correction. This value is represented in a 1.10 format with 1 integer and 10 fractional bits. The value should always be programmed to be less than 1.0.
1	31:11	**Reserved** Project: All Format: MBZ
	10:0	**GAMC16G** Project: All This value specifies the 17th reference point that is used for the green color channel sprite gamma correction. This value is represented in a 1.10 format with 1 integer and 10 fractional bits. The value should always be programmed to be less than 1.0.
2	31:11	**Reserved** Project: All Format: MBZ
	10:0	**GAMC16B** Project: All This value specifies the 17th reference point that is used for the blue color channel sprite gamma correction. This value is represented in a 1.10 format with 1 integer and 10 fractional bits. The value should always be programmed to be less than 1.0.

4.4.14 SPR_GAMC17—Sprite Gamma Correction Point 17

SPR_GAMC17	
Register Type:	**MMIO**
Project:	**All**
Default Value:	**00000BFFh; 00000BFFh; 00000BFFh;**
Access:	**R/W**
Size (in bits):	**3x32**

These registers are used to determine the 18th reference point (point 17 when counting from 0) for sprite gamma correction. See SPR_GAMC for sprite gamma programming information.

DWord	Bit	Description
0	31:12	**Reserved** Project: All Format: MBZ
	11:0	**GAMC17R** Project: All This value specifies the 18th reference point that is used for the red color channel sprite gamma correction. This value is represented in a 2.10 format with 2 integer and 10 fractional bits. The value should always be programmed to be less than 3.0.
1	31:12	**Reserved** Project: All Format: MBZ
	11:0	**GAMC17G** Project: All This value specifies the 18th reference point that is used for the green color channel sprite gamma correction. This value is represented in a 2.10 format with 2 integer and 10 fractional bits. The value should always be programmed to be less than 3.0.
2	31:12	**Reserved** Project: All Format: MBZ
	11:0	**GAMC17B** Project: All This value specifies the 18th reference point that is used for the blue color channel sprite gamma correction. This value is represented in a 2.10 format with 2 integer and 10 fractional bits. The value should always be programmed to be less than 3.0.