# Intel® OpenSource HD Graphics Programmer's Reference Manual (PRM)

## Volume 3 Part 2: Display Registers – CPU Registers (SandyBridge)

## For the 2011 Intel Core Processor Family

***May 2011***

***Revision 1.0***

***NOTICE:***

This document contains information on products in the design phase of development, and Intel reserves the right to add or remove product features at any time, with or without changes to this open source documentation.

*Doc Ref #: IHD-OS-V3 Pt2 – 05 11*

# Contents

# 1. CPU Display Registers [DevSNB+]

This chapter discusses CPU Display Registers that are active in SandyBridge and later projects, unless indicated otherwise.

## 1.1 1.1 Introduction and Register Summary

This chapter contains the register descriptions for the display portion of a family of integrated graphics devices. These registers do 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.

The following table contains the sections break down where the register information is contained within this chapter:

| Address Range | Description |
|---|---|
| 40000h-4FFFFh | Shared Functions |
| 60000h–6FFFFh | Pipe and Port Controls |
| 70000h–7FFFFh | Plane Controls |

### 1.1.1 Terminology

| Description | Software Use | Should be implemented as |
|---|---|---|
| Read/Write | 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: write as zero, 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 | 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. |

| Description | Software Use | Should be implemented as |
|---|---|---|
| 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.<br><br>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 needed for specific arming sequences. |

### 1.1.2 Display Mode Set Sequence

| Wait values |
|---|
| CPU DP PLL warmup = 20uS<br>CPU FDI transmitter PLL warmup = 10us<br>DMI latency = 20uS<br>FDI training pattern 1 time = 0.5uS<br>FDI training pattern 2 time = 1.5uS<br>FDI idle pattern time = 31uS |
| **Enable sequence** |
| 1. Enable panel power as needed to retrieve panel configuration (use AUX VDD enable bit<br>2. Enable PCH clock reference source and PCH SSC modulator, wait for warmup (Can be done anytime before enabling port)<br>3. If enabling CPU embedded DisplayPort A: (Can be done anytime before enabling CPU pipe or port)<br>a. Enable PCH 120MHz clock source output to CPU, wait for DMI latency<br>b. Configure and enable CPU DisplayPort PLL in the DisplayPort A register, wait for warmup<br>4. If enabling port on PCH: (Must be done before enabling CPU pipe or FDI)<br>a. Enable PCH FDI Receiver PLL, wait for warmup plus DMI latency<br>b. Switch from Rawclk to PCDclk in FDI Receiver (FDI A OR FDI B)<br>c. [DevSNB] Enable CPU FDI Transmitter PLL, wait for warmup<br>d. [DevILK] CPU FDI PLL is always on and does not need to be enabled<br>5. Enable CPU panel fitter if needed for hires, required for VGA (Can be done anytime before enabling CPU pipe)<br>6. Configure CPU pipe timings, M/N/TU, and other pipe settings (Can be done anytime before enabling CPU pipe)<br>7. Enable CPU pipe<br>8. Configure and enable CPU planes (VGA or hires)<br>9. If enabling port on PCH:<br>a. Program PCH FDI Receiver TU size same as Transmitter TU size for TU error checking<br>b. Train FDI<br>i. Set pre-emphasis and voltage (iterate if training steps fail)<br>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 ISR ([DevIBX-B+] 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 ISR ([DevIBX-B+] 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. [DevCPT] Configure DPLL SEL to set the DPLL to transcoder mapping and enable DPLL to the transcoder.
e. [DevCPT] Configure DPLL_CTL DPLL_HDMI_multipler.
f. Configure PCH transcoder timings, M/N/TU, and other transcoder settings (should match CPU settings).
g. [DevCPT] Configure and enable Transcoder DisplayPort Control if DisplayPort will be used
h. Enable PCH transcoder

10. Enable ports (DisplayPort must enable in training pattern 1)
11. Enable panel power through panel power sequencing
12. Wait for panel power sequencing to reach enabled steady state
13. Disable panel power override
14. If DisplayPort, complete link training
15. 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. [DevILK-A] Disable CPU panel fitter
5. Disable CPU pipe
6. Wait for CPU pipe off status (CPU pipe config register pipe state)
7. If disabling DisplayPort on PCH, write the DisplayPort control register bit 31 to 0b.
8. Disable CPU panel fitter (Can be done anytime after CPU pipe is off)
9. If disabling CPU embedded DisplayPort A
    a. Disable port
    b. Disable CPU DisplayPort PLL in the DisplayPort A register
    c. Disable PCH 120MHz clock source output to CPU
10. 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)<br>e. [DevCPT] Disable Transcoder DisplayPort Control if DisplayPort was used<br>f. [DevCPT] Disable Transcoder DPLL Enable bit in DPLL_SEL<br>g. Disable PCH DPLL (Can be done anytime after PCH ports and transcoder are off)<br>h. If no other PCH transcoder is enabled<br>i. Switch from PCDclk to Rawclk in PCH FDI Receiver<br>ii. [DevSNB] Disable CPU FDI Transmitter PLL<br>iii. Disable PCH FDI Receiver PLL<br>11. If SSC is no longer needed, disable PCH SSC modulator<br>12. 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.<br>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. |

# 2. North Shared Functions (40000h–4FFFFh)

## 2.1 VGA Control Registers

### VGACNTRL—VGA Display Plane Control Register

**Register Type:** MMIO
**Address Offset:** 41000h
**Project:** All
**Default Value:** 00002900h
**Access:** R/W
**Size (in bits):** 32

**Note**: VGA requires panel fitting to be enabled.

| Bit | Description |
|---|---|
| 31 | **VGA_Display_Disable**<br>Project: All<br>Default Value: 0b VGA Display Enabled<br>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 A) can be enabled. **Note**: The VGA SR01 screen off bit must be programmed when enabling and disabling VGA. See the VGA Registers Bspec.<br><br>Firmware can temporarily override VGA display to be disabled, causing this bit to become 1b (Disable). In order to properly disable VGA this bit should be programmed to 1b even if it already reads as 1b. |
| 30 | **Reserved** Project: All Format: PBC |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Enable | VGA Display Enabled | All |
| 1b | Disable | VGA Display Disabled | All |

## VGACNTRL—VGA Display Plane Control Register

**29** **VGA_Pipe_Select**

Project: All

Default Value: 0b

For dual pipe devices, this bit determines which pipe is to receive the VGA display data. This must be changed only when the VGA display is in the disabled state via the VGA display disable bit or during the write to enable VGA display.

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | PipeA | Selects Assigns the VGA display to Pipe A | All |
| 1b | PipeB | Selects Assigns the VGA display to Pipe B | All |

**28:27** **Reserved** Project: All Format: PBC

**26** **VGA_Border_Enable**

Project: All

Default Value: 0b

This bit determines if the VGA border areas are included in the active display area and do or do not appear on the port output.

The border if enabled will be scaled along with the pixel data. Setting this bit allows the popup to be positioned overlapping the border area of the image.

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Disable | VGA Border areas are not included in the image size calculations for active area. | All |
| 1b | Enable | VGA Border areas are enabled and passed to the display pipe for display and used in the image size calculations. | All |

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

**24** **Pipe_Color_Space_Conversion_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 |

## VGACNTRL—VGA Display Plane Control Register

23 **VGA_Palette_Read_Select**

Project: All

Default Value: 0b

This bit only applies to dual display pipe devices and determines which palette VGA palette read accesses will occur from.

VGA palette reads are reads from I/O address 0x3c9.

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Palette A | VGA palette reads will access Palette A | All |
| 1b | Palette B | VGA palette reads will access Palette B | All |

22 **VGA_Palette_A_Write_Disable**

Project: All

Default Value: 0b

This determines which palette the VGA palette writes will have as a destination. One or both palettes can be the destination. If both are disabled, writes will not affect the palette contents.

VGA palette writes are writes to I/O address 0x3C9h.

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Update Palette A | VGA palette writes will update Palette A | All |
| 1b | Not Update Palette A | VGA palette writes will not update Palette A | All |

21 **VGA_Palette_B_Write_Disable**

Project: All

Default Value: 0b

This determines which palette the VGA palette writes will have as a destination. One or both palettes can be the destination. If both are disabled, writes will not affect the palette contents.

VGA palette writes are writes to I/O address 0x3C9h.

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Update Palette B | VGA palette writes will update Palette B | All |
| 1b | Not Update Palette B | VGA palette writes will not update Palette B | All |

20 **Legacy_VGA_8-Bit_Palette_Enable**

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 |

## VGACNTRL—VGA Display Plane Control Register

| | |
|---|---|
| 19 | **Reserved**<br>Project: All |
| 18 | **Reserved** |
| 17:16 | **Reserved** Project: All |
| 15:12 | **Reserved** |
| 11:8 | Reserved |
| 7:6 | **Blink_Duty_Cycle**<br>Project: All<br>Default Value: 00b<br>Controls the VGA text mode blink duty cycle relative to the VGA cursor blink duty cycle. |
| 5:0 | **VSYNC_Blink_Rate** Project: All<br>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. |

| 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 |

## 2.2 Sine ROM Registers

### 2.2.1 2.4.1 SINE_ROM—Sine ROM

| SINE_ROM - Sine ROM | |
|---|---|
| **Register Type:** | MMIO |
| **Address:** | 42200h-42203h |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | R/W Special |
| **Size (in bits):** | 32 |

This register can be used to calculate a sine (or cosine). The intent is to enable calculation of filter coefficients. The angle is written to bits [16:6] as a 11 bit, fixed point, 0.11 value (example for setting different degrees below). Then the sine is read from bits [16:6] as a 11 bit, fixed point, 1.10 value.

| DWord | Bit | Description |
|---|---|---|
| 0 | 31:17 | **Reserved**<br>Project: All |
| | 16:6 | **Sine**<br>Default Value: 0b<br>Project: All<br>Write the angle, read the sine<br>**Programming Notes**: Examples of values to write:00000000000b = 0 or 360 degrees01000000000b = 90 degrees10000000000b = 180 degrees11000000000b = 270 degrees<br>**Project**: All |
| | 5:0 | **Reserved**<br>Project: All |

## 2.3 Power Measurement Registers

These registers are read by the PMU to get information for use in device power estimation.

### 2.3.1 DE_POWER1 – Display Engine Power Register 1

| DE_POWER1 – Display Engine Power Register 1 | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | 42400h-42403h |
| **Project:** | DevSNB |
| **Default Value:** | 00000000h |
| **Access:** | Read Only |
| **Size (in bits):** | 32 |

| Bit | Description |
|---|---|
| 31:8 | **Reserved** Project: All Format: MBZ |
| 7:4 | **Transmit_Lanes_Enabled**<br>Project: All<br>Range 0..12<br>The total number of eDP & FDI lanes enabled. |
| 3:2 | **Enabled_Panel_Fitters**<br>Project: All<br>Range 0..2<br>Each enabled panel fitter consumes an additional xxmW of power. |
| 1:0 | **Enabled_DPLLs**<br>Project: All<br>Range 0..2<br>Each DPLL enabled consumes xxmW of power. |

### 2.3.2 DE_POWER2 – Display Engine Power Register 2

| DE_POWER2 – Display Engine Power Register 2 | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **42404h-42407h** |
| **Project:** | DevSNB |
| **Default Value:** | 00000000h |
| **Access:** | Read Only |
| **Size (in bits):** | 32 |

| Bit | Description |
|---|---|
| 31:0 | **DE_bandwidth_counter** Project: All<br>This counter increments on every cache line put arriving at the DE. The bandwidth is estimated by taking the difference between two reads at a known interval. The counter is only reset at boot time. |

## 2.4 DPFC Control Registers (43200h–433FFh)

### 2.4.1 DPFC_CB_BASE – DPFC Compressed Buffer Base Address

| DPFC_CB_BASE – DPFC Compressed Buffer Base Address | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | 43200h-43203h |
| **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 | **Compressed_Frame_Buffer_Offset_Address** Project: All<br>This register specifies offset of the Compressed Frame Buffer from the base of stolen memory.<br>The buffer must be 4K byte aligned. |
| 11:0 | **Reserved** Project: All Format: MBZ |

## 2.4.2 DPFC_CONTROL— DPFC Control

### DPFC_CONTROL— DPFC Control

**Register Type:** MMIO
**Address Offset:** 43208h-4320Bh
**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

| Bit | Description |
|---|---|
| 31 | **Enable_Frame_Buffer_Compression**<br>Project: All<br>Default Value: 0b<br>This bit is used to globally enable DPFC function at the next Vertical Blank start.<br>Frame buffer compression can only be enabled after selected primary plane has been enabled for one or more vertical blanks and must be disabled before disabling the primary plane. |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Disable | Disable frame buffer compression | All |
| 1b | Enable | Enable frame buffer compression | All |

| Bit | Description |
|---|---|
| 30 | **Plane_Select**<br>Project: All<br>Default Value: 0b |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Plane A | Plane A | All |
| 1b | Plane B | Plane B | All |

| Bit | Description |
|---|---|
| 29 | **CPU_Fence_Enable**<br>Project: All<br>Default Value: 0b |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | No CPU Disp Buf | Display Buffer is not in a CPU fence. No modifications are allowed from CPU to the Display Buffer | All |
| 1b | CPU Disp Buf | Display Buffer exists in a CPU fence | All |

| Bit | Description |
|---|---|
| 28 | **Reserved** Project: All Format: MBZ |
| 27 | Reserved |
| 26 | **Reserved** |

## DPFC_CONTROL— DPFC Control

25 **Persistent_Mode**

Project: All

Default Value: 0b

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Non Persistent | Non Persistent Mode | All |
| 1b | Persistent | Persistent Mode. Enable the invalid modify qualify from CS | All |

24:8 **Reserved**

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 1, Pixel Format 16 bpp - Not Supported

Compression Ratio 1, Pixel Format 32 bpp - Supported (CFB=FB)

Compression Ratio 1/2, Pixel Format 16 bpp - Supported (CFB=FB)

Compression Ratio 1/2, Pixel Format 32 bpp - Supported (CFB=1/2 FB)

Compression Ratio 1/4, Pixel Format 16 bpp - Supported (CFB=1/2FB)

Compression Ratio 1/4, Pixel Format 32 bpp – Supported (CFB=1/4 FB)

FB = Frame Buffer Size

CFB = Compressed Frame Buffer Size

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | **[Default]** | | |
| 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 |

## DPFC_CONTROL— DPFC Control

| | |
|---|---|
| 5:4 | **Write Back Watermark**<br>Default Value: 0b<br>Project: All<br>Compressed data write back engine waits for this amount of data (per segment) to be ready before writing the data out to memory. Compression SR mode must be a 1, or SR disabled for this to take effect. |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | **[Default]** | | |
| 00b | 4 cache lines | 4 cache lines | All |
| 01b | 8 cache lines | 8 cache lines | All |
| 1Xb | Reserved | Reserved | All |

| | |
|---|---|
| 3:0 | **CPU_Fence_Number**<br>Project: All<br>Default Value: 0b<br>This field specifies the CPU visible FENCE number corresponding to the placement of the uncompressed frame buffer.<br>**[DevSNB]** 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. |

## 2.4.3 DPFC_RECOMP_CTL — DPFC ReComp Control

### DPFC_RECOMP_CTL — DPFC ReComp Control

**Register Type:** MMIO
**Address Offset:** 4320Ch-4320Fh
**Project:** All
**Default Value:** 00000000h
**Access:** R/W
**Size (in bits):** 32

| Bit | Description |
|---|---|
| 31:28 | **Reserved** Project: All Format: MBZ |
| 27 | **Enable_ReComp_Stall**<br>Project: All<br>Default Value: 0b |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Disable [Default] | Disable | All |
| 1b | Enable | Enable | All |

| DPFC_RECOMP_CTL — DPFC ReComp Control | |
|---|---|
| 26:16 | **ReComp_Stall_Invalidation_Watermark** Project: All<br>If this many or more invalidations occur in one frame, stop compression until the number falls below watermark, then start the recomp timer. |
| 15:6 | **Reserved** Project: All Format: MBZ |
| 5:0 | **ReCompression_Timer_Count** Project: All<br>After invalidations fall below watermark, wait this many frames before restarting the compressor.<br>A 0 means restart compression on the following frame. |

### 2.4.4 DPFC_STATUS — DPFC Status

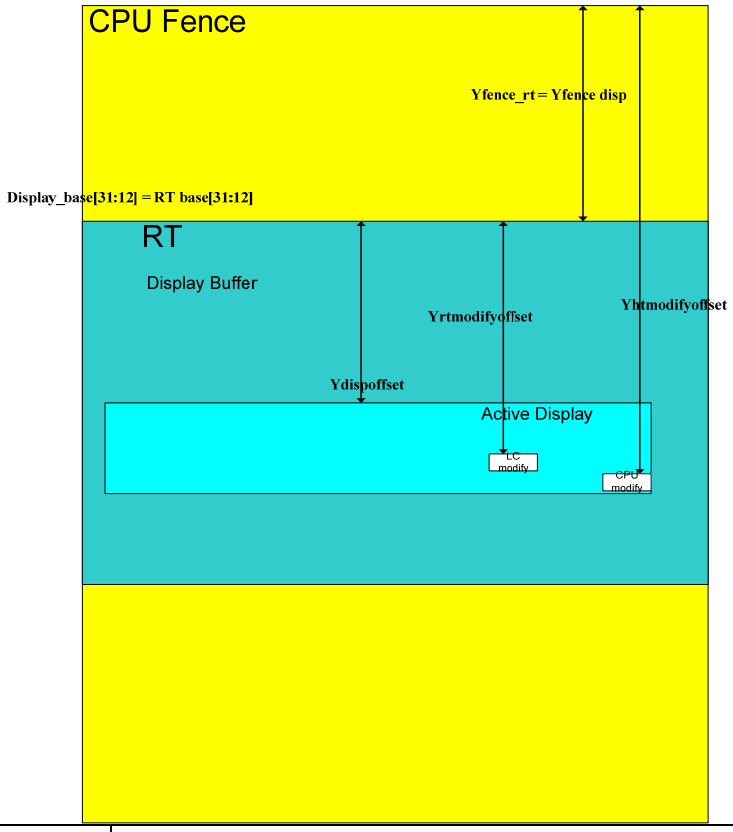
| DPFC_STATUS — DPFC Status | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | 43210h-43213h |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | Read Only |
| **Size (in bits):** | 32 |
| **Bit** | **Description** |
| 31:27 | **Reserved** Project: All Format: MBZ |
| 26:16 | **RC_Invalidated_Segment_Count** Project: [DevSNB]<br>Updated each vblank, this field indicates the number of segments that have been invalidated for the previous frame, for RC invalidation only |
| 15:11 | **Reserved** Project: All Format: MBZ |
| 10:0 | **Compressed_Segment_Count** Project: All<br>Updated each vblank, this field indicates the number of segments that were fetched from the compressed frame buffer for the previous frame. |

## 2.4.5 DPFC_CPU_Fence_Offset — Y Offset CPU Fence Base to Display Buffer Base

### DPFC_CPU_Fence_Offset — Y Offset CPU Fence Base to Display Buffer Base

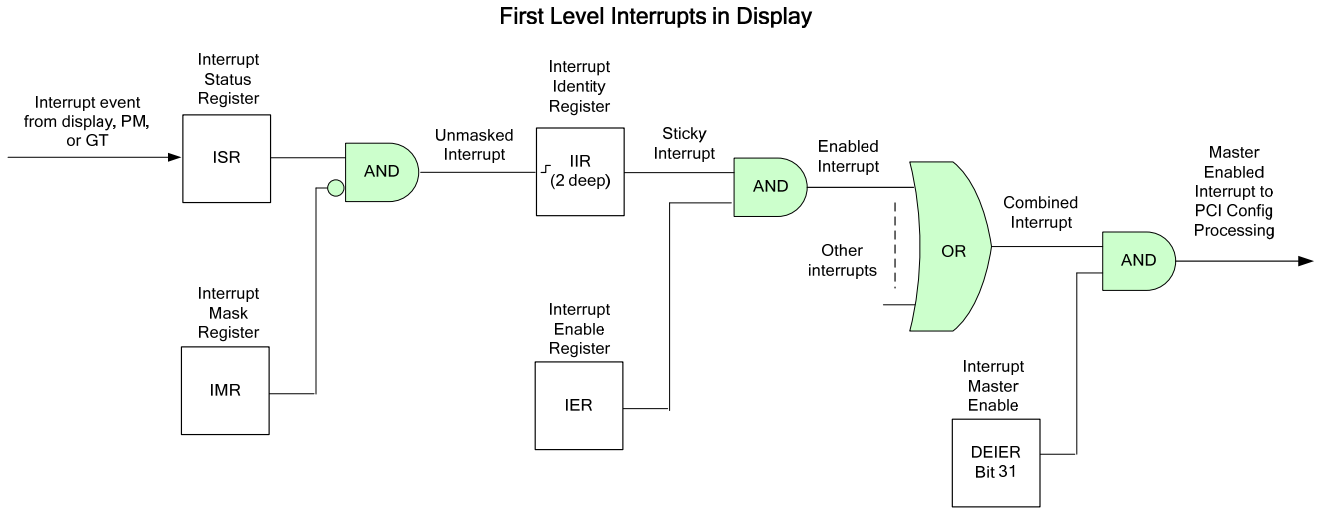
| | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | 43218h-4321Bh |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | R/W |
| **Size (in bits):** | 32 |

The contents of this register (shown below) can not be changed while compression is enabled.




| Bit | Description |
|---|---|
| 31:22 | **Reserved** Project: All Format: MBZ |
| 21:0 | **Yfence_disp** Project: All<br>Y offset from the CPU fence to the Display Buffer base.<br>**[DevSNB]** The CPU fence is always programmed to match the Display Buffer base, so this offset must be programmed to 0 to match. |

## 2.5 Interrupt Control Registers




For every first level interrupt bit:

Interrupt event comes in from display, PM, or GT.

There may be more levels of interrupt handling behind each event. For example the PCH Display interrupt event is the result of the SDE interrupt registers.

Interrupt event goes to Interrupt Status Register (ISR) where live status can be read back.

Interrupt event is ANDed with inverted Interrupt Mask Register (IMR) so only unmasked interrupts will proceed.

Unmasked interrupt rising edge sets sticky bit in Interrupt Indentity Register (IIR).

IIR is cleared by writing a 1b to it.

IIR can queue up to two interrupt events. When the IIR is cleared, it will set itself again after one clock if a second event was stored.

Sticky interrupt is ANDed with Interrupt Enable Register (IER) so only enabled interrupts will proceed.

All enabled interrupts are then ORed to create the combined interrupt.

Combined interrupt is ANDed with Master Enable (DEIER Bit 31) so only master enabled interrupt will proceed.

Master enabled interrupt then goes to PCI device 2 configuration registers PCISTS2, PCICMD2, and MC which control the MSI and line interrupt.

A Function Level Reset (FLR) or Warm Reset will reset all interrupt logic in display, causing the master enabled interrupt to deassert.

### 2.5.1 Display Engine Interrupt Registers Bit Definition

| Display Engine Interrupt Registers 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 DEIIR and GTIIR and PMIIR 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:<br>This bit exists only in the DEIER Display Engine Interrupt Enable Register.<br>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. |
| 30 | **Reserved** Project: All Format: |
| 29 | **Sprite_Plane_B_flip_done** Project: All Format:<br>This is an active high pulse when a sprite plane B flip is done. |
| 28 | **Sprite_Plane_A_flip_done** Project: All Format:<br>This is an active high pulse when a sprite plane A flip is done. |
| 27 | **Primary_Plane_B_flip_done** Project: All Format:<br>This is an active high pulse when a primary plane B flip is done. |
| 26 | **Primary_Plane_A_flip_done** Project: All Format:<br>This is an active high pulse when a primary plane A flip is done. |
| 25 | **Reserved** Project: **DevSNB** Format: |
| 24 | **GTT_fault** Project: All Format:<br>This is an active high level while either of the GTT Fault Status register bits are set. |
| 23 | **Poison** Project: All Format:<br>This is an active high pulse on receiving the poison message. |
| 21 | **PCH_Display_interrupt_event** Project: All Format:<br>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. Only the rising edge of the PCH Display interrupt will cause the IIR to be set here, so all PCH Display Interrupts, including back to back interrupts, must be cleared before a new PCH Display Interrupt can cause the IIR to be set here. |
| 20 | **AUX_Channel_A** Project: All Format:<br>This is an active high pulse on the AUX A done event. |
| 19 | **DP_A_Hotplug** Project: All Format:<br>This is an active high level while either of the Digital Port A Hot Plug Interrupt Detect Status register bits are set. |
| 18 | **GSE** Project: DevSNB Format:<br>This is an active high pulse on the GSE system level event. |

| Display Engine Interrupt Registers Bit Definition | |
|---|---|
| 17 | **DPST_histogram_event** Project: All Format:<br>This is an active high pulse on the DPST histogram event. |
| 16 | **DPST_phase_in_event** Project: All Format:<br>This is an active high pulse on the DPST phase in event. |
| 15 | **Pipe_B_vblank** Project: All Format:<br>This is an active high level for the duration of the Pipe B vertical blank. |
| 14 | **Pipe_B_even_field** Project: All Format:<br>This is an active high level for the duration of the Pipe B interlaced even field. |
| 13 | **Pipe_B_odd_field** Project: All Format:<br>This is an active high level for the duration of the Pipe B interlaced odd field. |
| 12 | **Pipe_B_line_compare** Project: All Format:<br>This is an active high level for the duration of the selected Pipe B scan lines. |
| 11 | **Pipe_B_vsync** Project: All Format:<br>This is an active high level for the duration of the Pipe B vertical sync. |
| 10 | **Pipe_B_CRC_done** Project: All Format:<br>This is an active high pulse on the Pipe B CRC done. |
| 9 | **Pipe_B_CRC_error** Project: All Format:<br>This is an active high pulse on the Pipe B CRC error. |
| 8 | **Pipe_B_FIFO_underrun** Project: All Format:<br>This is an active high level for the duration of the Pipe B FIFO underrun. |
| 7 | **Pipe_A_vblank** Project: All Format:<br>This is an active high level for the duration of the Pipe A vertical blank. |
| 6 | **Pipe_A_even_field** Project: All Format:<br>This is an active high level for the duration of the Pipe A interlaced even field. |
| 5 | **Pipe_A_odd_field** Project: All Format:<br>This is an active high level for the duration of the Pipe A interlaced odd field. |
| 4 | **Pipe_A_line_compare** Project: All Format:<br>This is an active high level for the duration of the selected Pipe A scan lines. |
| 3 | **Pipe_A_vsync** Project: All Format:<br>This is an active high level for the duration of the Pipe A vertical sync. |
| 2 | **Pipe_A_CRC_done** Project: All Format:<br>This is an active high pulse on the Pipe A CRC done. |
| 1 | **Pipe_A_CRC_error** Project: All Format:<br>This is an active high pulse on the Pipe A CRC error. |
| 0 | **Pipe_A_FIFO_underrun** Project: All Format:<br>This is an active high level for the duration of the Pipe A FIFO underrun. |

## 2.5.2 DEISR — Display Engine Interrupt Status Register

### DEISR — Display Engine Interrupt Status Register

**Register Type:** MMIO
**Address Offset:** 44000h-44003h
**Project:** All
**Default Value:** 00000000h
**Access:** Read Only
**Size (in bits):** 32

The ISR register contains the non-persistent value of all interrupt status bits. The IMR register selects which of these interrupt conditions are reported in the persistent IIR (i.e., set bits must be cleared by software). Bits in the IER are used to selectively enable IIR bits to cause CPU interrupts.

| Bit | Description |
|---|---|
| 31:0 | **Display_Engine_ Interrupt_ Status_Bits**<br>Project: All<br>Format: Display Engine Interrupt Registers Bit Definition<br>FormatDesc: Display Engine (DE) interrupt bits come from events within the display engine, except for some which explicitly list a non-display engine source. The DEIIR and GTIIR and PMIIR are ORed together to generate the CPU interrupt.<br>The Display Engine Interrupt Control Registers all share the same bit definitions from this table. |

| 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. |

### 2.5.3 DEIMR — Display Engine Interrupt Mask Register

| DEIMR — Display Engine Interrupt Mask Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | 44004h-44007h |
| **Project:** | All |
| **Default Value:** | FFFFFFFFh |
| **Access:** | R/W |
| **Size (in bits):** | 32 |

The IMR register is used by software to control which Interrupt Status Register bits are “masked” or “unmasked”. “Unmasked” bits will be reported in the IIR, possibly triggering a CPU interrupt, and will persist in the IIR until cleared by software. “Masked” bits will not be reported in the IIR and therefore cannot generate CPU interrupts.

| Bit | Description |
|---|---|
| 31:0 | **Display Engine Interrupt Mask Bits**<br>Project: All<br>Format: Display Engine Interrupt Registers Bit Definition<br>FormatDesc: Display Engine (DE) interrupt bits come from events within the display engine, except for some which explicitly list a non-display engine source. The DEIIR and GTIIR and PMIIR are ORed together to generate the CPU interrupt.<br>The Display Engine Interrupt Control Registers all share the same bit definitions from this table.<br>This field contains a bit mask which selects which interrupt bits from the ISR are reported in the IIR. |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Not Masked | Not Masked – will be reported in the IIR | All |
| 1b | Masked | Masked – will not be reported in the IIR | All |

## 2.5.4 DEIIR — Display Engine Interrupt Identity Register

### DEIIR — Display Engine Interrupt Identity Register

**Register Type:** MMIO
**Address Offset:** 44008h
**Project:** All
**Default Value:** 00000000h
**Access:** R/W Clear
**Size (in bits):** 32

The IIR register contains the interrupt bits that are “unmasked” by the IMR and thus can generate CPU interrupts (if enabled via the IER). When a CPU interrupt is generated, this should be the first register to be interrogated to determine the source of the interrupt. **Writing a ‘1’ into the appropriate bit position within this register clears interrupts.**

| Bit | Description |
|---|---|
| 31:0 | **Display Engine Interrupt Identity Bits**<br>Project: All<br>Format: Display Engine Interrupt Registers Bit Definition<br>FormatDesc: Display Engine (DE) interrupt bits come from events within the display engine, except for some which explicitly list a non-display engine source. The DEIIR and GTIIR and PMIIR are ORed together to generate the CPU interrupt.<br>The Display Engine Interrupt Control Registers all share the same bit definitions from this table.<br>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 via software by writing a 1' to the appropriate bit(s).<br>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 actually generated a CPU interrupt) | All |

## 2.5.5 DEIER — Display Engine Interrupt Enable Register

### DEIER — Display Engine Interrupt Enable Register

**Register Type:** MMIO
**Address Offset:** 4400Ch
**Project:** All
**Default Value:** 00000000h
**Access:** R/W
**Size (in bits):** 32

The IER register contains an interrupt enable bit for each interrupt bit in the IIR register. A disabled interrupt will still appear in the Interrupt Identity Register to allow polling of interrupt sources.

| Bit | Description |
|---|---|
| 31:0 | **Display Engine Interrupt Enable Bits**<br>Project: All<br>Format: Display Engine Interrupt Registers Bit Definition<br>FormatDesc: Display Engine (DE) interrupt bits come from events within the display engine, except for some which explicitly list a non-display engine source. The DEIIR and GTIIR and PMIIR are ORed together to generate the CPU interrupt.<br>The Display Engine Interrupt Control Registers all share the same bit definitions from this table.<br>The bits in this register enable a CPU interrupt to be generated whenever the corresponding bit in the Interrupt Identity Register becomes set. The DEIER master interrupt control bit must be set to 1 for any interrupts to propagate to the system. |

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

## 2.5.6 GT Interrupt Registers Bit Definition

| GT Interrupt Registers Bit Definition | |
|---|---|
| **Project:** | All |
| **Size (in bits):** | 32 |
| The DEIIR and GTIIR and PMIIR are ORed together to generate the CPU interrupt.<br>The GT Interrupt Control Registers all share the same bit definition from this table. | |
| **Bit** | **Description** |
| 31 | **Reserved** Project: DevSNB Format: |
| 30 | **Blitter_AS_Context_Switch_Interrupt**<br>Project: DevSNB |
| 29 | **Blitter_page_directory_faults**<br>Project: DevSNB |
| 28:27 | **Reserved** Project: DevSNB Format: |
| 26 | **Blitter_MI_FLUSH_DW_notify**<br>Project: DevSNB |
| 25 | **Blitter_Command_Streamer_error_interrupt**<br>Project: DevSNB |
| 24 | **Billter_MMIO_sync_flush_status**<br>Project: DevSNB |
| 23 | **Reserved** Project: DevSNB Format: |
| 22 | **Blitter_Command_Streamer_MI_USER_INTERRUPT**<br>Project: DevSNB |
| 21 | **Reserved** Project: DevSNB Format: |
| 20 | **Video_AS_Context_Switch_Interrupt**<br>Project: DevSNB |

| GT Interrupt Registers Bit Definition | | | | | |
|---|---|---|---|---|---|
| 19 | **Video_page_directory_faults**<br>Project: DevSNB | | | | |
| 18 | **Video_Command_Streamer_Watchdog_counter_exceeded**<br>Project: DevSNB | | | | |
| 17 | **Reserved** | Project: | DevSNB | Format: | MBZ |
| 16 | **Video_MI_FLUSH_DW_notify**<br>Project: DevSNB | | | | |
| 15 | **Video_Command_Streamer_error_interrupt**<br>Project: DevSNB | | | | |
| 14 | **Video_MMIO_sync_flush_status**<br>Project: DevSNB | | | | |
| 13 | **Reserved** | Project: | DevSNB | Format: | |
| 12 | **Video_Command_Streamer_MI_USER_INTERRUPT**<br>Project: DevSNB | | | | |
| 11:9 | **Reserved** | Project: | DevSNB | Format: | |
| 8 | **Render_AS_Context_Switch_Interrupt**<br>Project: DevSNB | | | | |
| 7 | **Render_page_directory_faults**<br>Project: DevSNB | | | | |
| 6 | **Render_Command_Streamer_Watchdog_counter_exceeded**<br>Project: DevSNB | | | | |
| 5 | **Reserved** | Project: | DevSNB | Format: | |
| 4 | **Render_PIPE_CONTROL_notify**<br>Project: DevSNB | | | | |
| 3 | **Render_Command_Streamer_error_interrupt**<br>Project: DevSNB | | | | |
| 2 | **Render_MMIO_sync_flush_status**<br>Project: DevSNB | | | | |

| GT Interrupt Registers Bit Definition | |
|---|---|
| 1 | **Reserved**<br>Project: DevSNB |
| 0 | **Render_Command_Streamer_MI_USER_INTERRUPT**<br>Project: DevSNB |

## 2.5.7 GTISR — GT Interrupt Status Register

### GTISR — GT Interrupt Status Register

**Register Type:** MMIO
**Address Offset:** 44010h
**Project:** All
**Default Value:** 00000000h
**Access:** Read Only
**Size (in bits):** 32

The ISR register contains the non-persistent value of all interrupt status bits. The IMR register selects which of these interrupt conditions are reported in the persistent IIR (i.e., set bits must be cleared by software). Bits in the IER are used to selectively enable IIR bits to cause CPU interrupts.

| Bit | Description |
|---|---|
| 31:0 | **GT_Interrupt_Status_Bits**<br>Project: All<br>Format: GT Interrupt Registers Bit Definition<br><br>Format Description: The DEIIR and GTIIR and PMIIR are ORed together to generate the CPU interrupt.<br><br>The GT Interrupt Control Registers all share the same bit definition from this table.<br>This field contains the non-persistent values of all interrupt status bits. |

| 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. |

## 2.5.8 GTIMR — GT Interrupt Mask Register

### GTIMR — GT Interrupt Mask Register

**Register Type:** MMIO
**Address Offset:** 44014h
**Project:** All
**Default Value:** FFFFFFFFh
**Access:** R/W
**Size (in bits):** 32

The IMR register is used by software to control which Interrupt Status Register bits are “masked” or “unmasked”. “Unmasked” bits will be reported in the IIR, possibly triggering a CPU interrupt, and will persist in the IIR until cleared by software. “Masked” bits will not be reported in the IIR and therefore cannot generate CPU interrupts.

For command streamer interrupts DO NOT use this register to mask interrupt events. Instead use the individual command streamer MASK bits.

| Bit | Description |
|---|---|
| 31:0 | **GT Interrupt Mask Bits**<br>Project: All<br>Format: GT Interrupt Registers Bit Definition<br>FormatDesc: The DEIIR and GTIIR and PMIIR are ORed together to generate the CPU interrupt.<br>The GT Interrupt Control Registers all share the same bit definition from this table.<br>This field contains a bit mask which selects which interrupt bits from the ISR are reported in the IIR. |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Not Masked | Not Masked – will be reported in the IIR | All |
| 1b | Masked | Masked – will not be reported in the IIR | All |

## 2.5.9 GTIIR — GT Interrupt Identity Register

### GTIIR - GT Interrupt Identity Register

| | |
|---|---|
| **Register Type:** | MMIO |
| **Address:** | 44018h-4401Bh |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | R/W Clear |
| **Size (in bits):** | 32 |

The IIR register contains the interrupt bits that are unmasked by the IMR and thus can generate CPU interrupts (if enabled via the IER). When a CPU interrupt is generated, this should be the first register to be interrogated to determine the source of the interrupt. **Writing a 1' into the appropriate bit position within this register clears interrupts.**

| DWord | Bit | Description |
|---|---|---|
| 0 | 31:0 | **GT Interrupt Identity Bits**<br>Project: All<br>Format: GT Interrupt Registers Bit Definition<br>FormatDesc: The DEIIR and GTIIR and PMIIR are ORed together to generate the CPU interrupt.<br>The GT Interrupt Control Registers all share the same bit definition from this table.<br>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 via software by writing a 1' to the appropriate bit(s).<br>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 actually generated a CPU interrupt) | All |

## 2.5.10 GTIER — GT Interrupt Enable Register

### GTIER - GT Interrupt Enable Register

| | |
|---|---|
| **Register Type:** | MMIO |
| **Address:** | 4401Ch-4401Fh |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | R/W |
| **Size (in bits):** | 32 |

The IER register contains an interrupt enable bit for each interrupt bit in the IIR register. A disabled interrupt will still appear in the Interrupt Identity Register to allow polling of interrupt sources.

| DWord | Bit | Description |
|---|---|---|
| 0 | 31:0 | **GT Interrupt Enable Bits**<br>Project: All<br>Format: GT Interrupt Registers Bit Definition<br>FormatDesc:<br>The DEIIR and GTIIR and PMIIR are ORed together to generate the CPU interrupt.<br>The GT Interrupt Control Registers all share the same bit definition from this table.<br>The bits in this register enable a CPU interrupt to be generated whenever the corresponding bit in the Interrupt Identity Register becomes set. The DEIER master interrupt control bit must be set to 1 for any interrupts to propagate to the system. |

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

## 2.5.11 Power Management Interrupt Registers Bit Definition [DevSNB]

### Power Management Interrupt Registers Bit Definition [DevSNB]

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

The DEIIR and GTIIR and PMIIR are ORed together to generate the CPU interrupt.

The Power Management Interrupt Control Registers all share the same bit definition from this table.

| Bit | Description |
|---|---|
| 31:26 | **Reserved** Project: All Format: |
| 25 | **PCU_pcode2driver_mailbox_event** Project: All Format: |
| 24 | **PCU_Thermal_Event** Project: All Format: |
| 23:7 | **Reserved** Project: All Format: |
| 6 | **Render_Frequency_Downward_Timeout_During_RC6_interrupt** Project: All Format: |
| 5 | **RP_UP_threshold_interrupt** Project: All Format: |
| 4 | **RP_DOWN_threshold_interrupt** Project: All Format: |
| 3 | **Reserved** Project: All Format: |
| 2 | **Render_geyserville_UP_evaluation_interval_interrupt** Project: All Format: |
| 1 | **Render_geyserville_Down_evaluation_interval_interrupt** Project: All Format: |
| 0 | **Reserved** Project: All Format: MBZ |

## 2.5.12 PMISR — PM Interrupt Status Register

### PMISR — PM Interrupt Status Register

**Register Type:** MMIO
**Address Offset:** 44020h
**Project:** DevSNB
**Default Value:** 00000000h
**Access:** Read Only
**Size (in bits):** 32

The ISR register contains the non-persistent value of all interrupt status bits. The IMR register selects which of these interrupt conditions are reported in the persistent IIR (i.e., set bits must be cleared by software). Bits in the IER are used to selectively enable IIR bits to cause CPU interrupts.

| Bit | Description |
|---|---|
| 31:0 | **Power Management Interrupt Status Bits**<br>Project: All<br>Format: Power Management Interrupt Registers Bit Definition<br>FormatDesc: The DEIIR and GTIIR and PMIIR are ORed together to generate the CPU interrupt.<br>The Power Management Interrupt Control Registers all share the same bit definition from this table.<br>This field contains the non-persistent values of all interrupt status bits. |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Condition Does Not Exist | Interrupt Condition currently does not exist | All |
| 1b | Condition Exists | Interrupt Condition currently exists | All |

| Programming Notes | Project |
|---|---|
| Some inputs to this register are short pulses, therefore software should not expect to use this register to sample these conditions. | All |

## 2.5.13 PMIMR — Power Management Interrupt Mask Register

### PMIMR — Power Management Interrupt Mask Register

**Register Type:** MMIO
**Address Offset:** 44024h-44027h
**Project:** DevSNB
**Default Value:** FFFFFFFFh
**Access:** R/W
**Size (in bits):** 32

The IMR register is used by software to control which Interrupt Status Register bits are “masked” or “unmasked”. “Unmasked” bits will be reported in the IIR, possibly triggering a CPU interrupt, and will persist in the IIR until cleared by software. “Masked” bits will not be reported in the IIR and therefore cannot generate CPU interrupts.

For power management interrupts DO NOT use this register to mask interrupt events. Instead use the individual power management MASK bits in the corresponding PMunit register space.

| Bit | Description |
|---|---|
| 31:0 | **Power Management Interrupt Mask Bits**<br>Project: All<br>Format: Power Management Interrupt Registers Bit Definition<br>FormatDesc: The DEIIR and GTIIR and PMIIR are ORed together to generate the CPU interrupt.<br>The Power Management Interrupt Control Registers all share the same bit definition from this table.<br>This field contains a bit mask which selects which interrupt bits from the ISR are reported in the IIR. |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Not Masked | Not Masked – will be reported in the IIR | All |
| 1b | Masked | Masked – will not be reported in the IIR | All |

## 2.5.14 PMIIR — Power Management Interrupt Identity Register

### PMIIR — Power Management Interrupt Identity Register

**Register Type:** MMIO
**Address Offset:** 44028h
**Project:** DevSNB
**Default Value:** 00000000h
**Access:** R/W Clear
**Size (in bits):** 32

The IIR register contains the interrupt bits that are “unmasked” by the IMR and thus can generate CPU interrupts (if enabled via the IER). When a CPU interrupt is generated, this should be the first register to be interrogated to determine the source of the interrupt. **Writing a ‘1’ into the appropriate bit position within this register clears interrupts**.

| Bit | Description |
|---|---|
| 31:0 | **Power Management Interrupt Identity Bits**<br>Project: All<br>Format: Power Management Interrupt Registers Bit Definition<br>FormatDesc: The DEIIR and GTIIR and PMIIR are ORed together to generate the CPU interrupt.<br>The Power Management Interrupt Control Registers all share the same bit definition from this table.<br>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 via software by writing a 1' to the appropriate bit(s).<br>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 | IC No Detect | Interrupt Condition Not Detected | All |
| 1b | IC Detect | Interrupt Condition Detected (may or may not have actually generated a CPU interrupt) | All |

## 2.5.15 PMIER — Power Management Interrupt Enable Register

### PMIER — Power Management Interrupt Enable Register

**Register Type:** MMIO
**Address Offset:** **4402Ch-4402Fh**
**Project:** DevSNB
**Default Value:** 00000000h
**Access:** R/W
**Size (in bits):** 32

The IER register contains an interrupt enable bit for each interrupt bit in the IIR register. A disabled interrupt will still appear in the Interrupt Identity Register to allow polling of interrupt sources.

| Bit | Description |
|---|---|
| 31:0 | **Power Management Interrupt Enable Bits**<br>Project: All<br>Format: Power Management Interrupt Registers Bit Definition<br>FormatDesc: The DEIIR and GTIIR and PMIIR are ORed together to generate the CPU interrupt.<br>The Power Management Interrupt Control Registers all share the same bit definition from this table.<br>The bits in this register enable a CPU interrupt to be generated whenever the corresponding bit in the Interrupt Identity Register becomes set. The DEIER master interrupt control bit must be set to 1 for any interrupts to propagate to the system. |

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

## 2.5.16 Digital Port Hot Plug Control Register

### Digital Port Hot Plug Control Register

**Register Type:** MMIO
**Address Offset:** **44030h-44033h**
**Project:** All
**Default Value:** 00000000h
**Access:** R/W
**Size (in bits):** 32

| Bit | Description |
|---|---|
| 31:5 | **Reserved** Project: All Format: |
| 4 | **Digital_Port_A_Hot_Plug_Detect_Input_Enable**<br>Project: All<br>Default Value: 0b<br>Controls the state of the HPD buffer for the digital port. The buffer state is independent of whether the port is enabled or not. |
| 3:2 | **Digital_Port_A_Hot_Plug_Short_Pulse_Duration**<br>Project: All<br>Default Value: 0b<br>These bits define the duration of the pulse defined as a short pulse. |
| 1:0 | **Digital_Port_A_Hot_Plug_Interrupt_Detect_Status**<br>Project: All<br>Access: R/W Clear<br>Default Value: 0b<br>This reflects hot plug detect status on the digital port. This bit is used for either monitor hotplug/unplug or for notification of a sink event. When either a long or short pulse is detected, one of these bits will set. These bits are ORed together to go to the main ISR hotplug register bit. These are sticky bits, cleared by writing 1s to them. |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Disable | Buffer disabled | All |
| 1b | Enable | Buffer enabled. Hot plugs bit reflect the electrical state of the HPD pin | All |

| Value | Name | Description | Project |
|---|---|---|---|
| 00b | 2ms | 2 ms | All |
| 01b | 4.5ms | 4.5 ms | All |
| 10b | 6ms | 6 ms | All |
| 11b | 100ms | 100 ms | All |

| Value | Name | Description | Project |
|---|---|---|---|
| 00b | Not Detected | Digital port hot plug event not detected | All |

| Digital Port Hot Plug Control Register | | | | |
|---|---|---|---|---|
| | 1Xb | Long Pulse | Digital port long pulse hot plug event detected | All |
| | X1b | Short Pulse | Digital port short pulse hot plug event detected | All |

## 2.5.17 GTT Fault Status Register

**GTT Fault Status Register**

**Register Type:** MMIO
**Address Offset:** **44040h-44043h**
**Project:** All
**Default Value:** 00000000h
**Access:** R/W Clear
**Size (in bits):** 32

| Bit | Description |
|---|---|
| 31:8 | **Reserved** Project: All Format: |
| 7 | **Invalid_GTT_page_table_entry**<br>Project: All<br>Default Value: 0b<br>This is a sticky bit, cleared by writing 1 to it. All the GTT Fault Status bits are ORed together to go to the main ISR GTT Fault bit. |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Not Detected | Event not detected | All |
| 1b | Detected | Event detected | All |

| Bit | Description |
|---|---|
| 6 | **Invalid_page_table_entry_data**<br>Project: All<br>Default Value: 0b<br>This is a sticky bit, cleared by writing 1 to it. All the GTT Fault Status bits are ORed together to go to the main ISR GTT Fault bit. |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Not Detected | Event not detected | All |
| 1b | Detected | Event detected | All |

## GTT Fault Status Register

5 **Cursor_B_GTT_Fault_Status**

Project: All

Default Value: 0b

This reflects GTT fault status for this plane. This is a sticky bit, cleared by writing 1 to it. All the GTT Fault Status bits are ORed together to go to the main ISR GTT Fault bit.

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Not Detected | Event not detected | All |
| 1b | Detected | Event detected | All |

4 **Cursor_A_GTT_Fault_Status**

Project: All

Default Value: 0b

This reflects GTT fault status for this plane. This is a sticky bit, cleared by writing 1 to it. All the GTT Fault Status bits are ORed together to go to the main ISR GTT Fault bit.

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Not Detected | Event not detected | All |
| 1b | Detected | Event detected | All |

3 **Sprite_B_GTT_Fault_Status**

Project: All

Default Value: 0b

This reflects GTT fault status for this plane. This is a sticky bit, cleared by writing 1 to it. All the GTT Fault Status bits are ORed together to go to the main ISR GTT Fault bit.

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Not Detected | Event not detected | All |
| 1b | Detected | Event detected | All |

2 **Sprite_A_GTT_Fault_Status**

Project: All

Default Value: 0b

This reflects GTT fault status for this plane. This is a sticky bit, cleared by writing 1 to it. All the GTT Fault Status bits are ORed together to go to the main ISR GTT Fault bit.

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Not Detected | Event not detected | All |
| 1b | Detected | Event detected | All |

| GTT Fault Status Register | |
|---|---|
| 1 | **Primary_B_GTT_Fault_Status**<br>Project: All<br>Default Value: 0b<br>This reflects GTT fault status for this plane. This is a sticky bit, cleared by writing 1 to it. All the GTT Fault Status bits are ORed together to go to the main ISR GTT Fault bit. |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Not Detected | Event not detected | All |
| 1b | Detected | Event detected | All |

| | |
|---|---|
| 0 | **Primary_A_GTT_Fault_Status**<br>Project: All<br>Default Value: 0b<br>This reflects GTT fault status for this plane. This is a sticky bit, cleared by writing 1 to it. All the GTT Fault Status bits are ORed together to go to the main ISR GTT Fault bit. |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Not Detected | Event not detected | All |
| 1b | Detected | Event detected | All |

## 2.6 Display Engine Render Response

Do not cause more than one display event to be reported in the render response.

Either mask off all but one event, using the Display Engine Render Response Mask Register (DERRMR 0x44050), or never initiate more than one event.

### 2.6.1 Display Engine Render Response Message Bit Definition

**Display Engine Render Response Message Bit Definition**

**Project:** DevSNB
**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 |
|---|---|
| 31:14 | **Reserved** Project: All Format: |
| 13 | **Pipe_B_Start_of_Horizontal_Blank_Event** Project: All Format:<br>This event will be reported on the start of the Pipe B Horizontal Blank. |
| 12 | **Reserved** Project: All Format: |

| Display Engine Render Response Message Bit Definition | |
|---|---|
| 11 | **Pipe_B_Start_of_Vertical_Blank_Event** Project: All Format:<br>This event will be reported on the start of the Pipe B Vertical Blank. |
| 10 | **Pipe_B_Sprite_Plane_Flip_Done_Event** Project: All Format:<br>This event 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:<br>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:<br>This event will be reported on the Pipe B scan line event. |
| 7:6 | **Reserved** Project: All Format: |
| 5 | **Pipe_A_Start_of_Horizontal_Blank_Event** Project: All Format:<br>This event will be reported on the start of the Pipe A Horizontal Blank. |
| 4 | **Reserved** Project: All Format: |
| 3 | **Pipe_A_Start_of_Vertical_Blank_Event** Project: All Format:<br>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:<br>This event 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:<br>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:<br>This event will be reported on the Pipe A Display scan line event. |

## 2.6.2 DERRMR — Display Engine Render Response Mask Register

### DERRMR - Display Engine Render Response Mask Register

| | |
|---|---|
| **Register Type:** | MMIO |
| **Address:** | 44050h-44053h |
| **Project:** | DevSNB |
| **Default Value:** | 0000FFFFh |
| **Access:** | R/W |
| **Size (in bits):** | 32 |

This register is used by software to control which render response message bits are masked or unmasked. Unmasked bits will cause a render response message to be sent and will be reported in that message. Masked bits will not be reported and will not cause a render response message to be sent.

Unmasked events will wake render (command streamer) as they occur, so for improved power savings it is recommended to only unmask events that are required. Programming this register can be done through MMIO or a command streamer LOAD_REGISTER_IMMEDIATE command.

| DWord | Bit | Description |
|---|---|---|
| 0 | 31:0 | **Display Engine Render Response Message Mask Bits**<br>Project: All<br>Format: Display Engine Render Response Message Bit Definition<br>FormatDesc: 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.<br>This field contains a bit mask which selects which events cause and are reported in the render response message. |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Not Masked | Not Masked – will be cause and be reported in the message | All |
| 1b | Masked | Masked – will not cause or be reported in the message | All |

## 2.7 Display Arbitration Control

### 2.7.1 DISP_ARB_CTL—Display Arbiter Control

**DISP_ARB_CTL—Display Arbiter Control**

**Register Type:** MMIO
**Address Offset:** **45000h-45003h**
**Project:** All
**Default Value:** C2240622h
**Access:** R/W
**Size (in bits):** 32
**Trusted Type:** 1

| Bit | Description |
|---|---|
| 31 | **Reserved** |
| 30 | **Reserved** |
| 29 | **Reserved** |
| 28 | **Reserved** Project: All Format: |
| 27:26 | **HP_Queue_Watermark**<br>Project: All<br>Default Value: 00b |
| 25:24 | **LP_Write_Request_Limit**<br>Project: All<br>Default Value: 10b 4<br>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. |
| 23:20 | **TLB_Request_Limit**<br>Project: All<br>Default Value: 0010b 2<br>Range: 1..15<br>The value in this register indicates the maximum number of TLB requests that can be made in an arbitration loop. Range 1 – 15, (default 2). Zero is not a valid programming. |

| Value | Name | Description | Project |
|---|---|---|---|
| 00b | 1 | 1 | All |
| 01b | 2 | 2 | All |
| 10b | 4 | 4 (default) | All |
| 11b | 8 | 8 | All |

## DISP_ARB_CTL—Display Arbiter Control

**19:16 TLB_Request_In-Flight_Limit**

Project: All

Default Value: 0100b 4

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. Range 1 – 15, (default 4). Zero is not a valid programming.

**15 Reserved**

**14:13 Address_Swizzling_for_Tiled-Surfaces**

Project: All

Default Value: 00b

DRAM configuration registers show if memory address swizzling is needed.

| Value | Name | Description | Project |
|---|---|---|---|
| 00b | No Display | No display request address swizzling | All |
| 01b | Enable | Enable display request address bit[6] swizzling for tiled surfaces | All |
| 1Xb | Reserved | Reserved | All |

**12 Reserved** Project: All Format:

**11:8 HP_Page_Break_Limit**

Project: All

Default Value: 0110b 6

Range: 1..15

The value in this register represents the maximum number of page breaks allowed in a HP request chain. Range 1 – 15, (default 6). Zero is not a valid programming.

**7 Reserved** Project: All Format:

**6:0 HP_Data_Request_Limit**

Project: All

Default Value: 01000010b 34

Range: 1..127

The value in this register represents the maximum number of cachelines allowed in a HP request chain. Range 1 – 127, (default 34). Zero is not a valid programming.

## 2.7.2 DISP_ARB_CTL2—Display Arbiter Control 2 [DevSNB]

### DISP_ARB_CTL2—Display Arbiter Control 2 [DevSNB]

**Register Type:** MMIO
**Address Offset:** **45004h-45007h**
**Project:** DevSNB
**Default Value:** 00000000h
**Access:** R/W
**Size (in bits):** 32

| Bit | Description |
|---|---|
| 31 | **Reserved** |
| 30:9 | **Reserved** Project: All Format: MBZ |
| 8 | **Fetch_Timing**<br>Project: All<br>Default Value: 0b<br>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. |
| 7 | **Opportunistic_Fetch_Behavior**<br>Project: All<br>Default Value: 0h<br>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. |
| 6 | **Reserved** Project: All Format: MBZ |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | FE inSR | Fetch on falling edge of inSR | All |
| 1b | Not inSR | Fetch when not inSR | All |

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

<table>
<tr><th colspan="2">DISP_ARB_CTL2—Display Arbiter Control 2 [DevSNB]</th></tr>
<tr><td>5:4</td><td>
Inflight_HP_Read_Request_Limit<br>
Project: All<br>
Default Value: 00b<br>
The value in this register represents the maximum number of HP read request transactions that can inflight at any given time.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>00b</td><td>128 HP</td><td>128 HP inflight transactions limit</td><td>All</td></tr>
<tr><td>01b</td><td>64 HP</td><td>64 HP inflight transactions limit</td><td>All</td></tr>
<tr><td>10b</td><td>32 HP</td><td>32 HP inflight transactions limit</td><td>All</td></tr>
<tr><td>11b</td><td>16 HP</td><td>16 HP inflight transactions limit</td><td>All</td></tr>
</table>
</td></tr>
<tr><td>3:2</td><td>Reserved Project: All Format:</td></tr>
<tr><td>1:0</td><td>
RTID_FIFO_Watermark<br>
Project: All<br>
Default Value: 0b<br>
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
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>00b</td><td>8 RTIDs</td><td>8 RTIDs available in FIFO</td><td>All</td></tr>
<tr><td>01b</td><td>16 RTIDs</td><td>16 RTIDs available in FIFO</td><td>All</td></tr>
<tr><td>10b</td><td>32 RTIDs</td><td>32 RTIDs available in FIFO</td><td>All</td></tr>
<tr><td>11b</td><td>RESERVED</td><td>Reserved</td><td>All</td></tr>
</table>
</td></tr>
</table>

## 2.8 Display Watermark Registers

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 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 WM2 is disabled, WM3 must also be disabled, and if WM1 is disabled, both WM2 and WM3 must also be disabled.

### 2.8.1 WM0_PIPE_A—Pipe A Main Watermarks

| WM0_PIPE_A—Pipe A Main Watermarks | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **45100h-45103h** |
| **Project:** | All |
| **Default Value:** | 00783818h |
| **Access:** | R/W |
| **Size (in bits):** | 32 |

| Bit | Description |
|---|---|
| 31:23 | **Reserved** Project: All Format: |
| 22:16 | **Pipe_A_Primary_Watermark** Project: All<br>Number in 64Bs of data in FIFO below which the Pipe A Primary stream will generate requests to memory |
| 15:14 | **Reserved** Project: All Format: |
| 13:8 | **Pipe_A_Sprite_Watermark** Project: All<br>Number in 64Bs of data in FIFO below which the Pipe A Sprite stream will generate requests to memory |
| 7:5 | **Reserved** Project: All Format: |
| 4:0 | **Pipe_A_Cursor_Watermark** Project: All<br>Number in 64Bs of data in FIFO below which the Pipe A Cursor stream will generate requests to memory |

### 2.8.2 WM0_PIPE_B—Pipe B Main Watermarks

| WM0_PIPE_B—Pipe B Main Watermarks | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **45104h-45107h** |
| **Project:** | All |
| **Default Value:** | 00783818h |
| **Access:** | R/W |
| **Size (in bits):** | 32 |
| **Trusted Type:** | 1 |

| Bit | Description |
|---|---|
| 31:23 | **Reserved** Project: All Format: |
| 22:16 | **Pipe_B_Primary_Watermark** Project: All<br>Number in 64Bs of data in FIFO below which the Pipe B Primary stream will generate requests to memory |
| 15:14 | **Reserved** Project: All Format: |

| WM0_PIPE_B—Pipe B Main Watermarks | |
|---|---|
| 13:8 | **Pipe_B_Sprite_Watermark** Project: All<br>Number in 64Bs of data in FIFO below which the Pipe B Sprite stream will generate requests to memory |
| 7:5 | **Reserved** Project: All Format: |
| 4:0 | **Pipe_B_Cursor_Watermark** Project: All<br>Number in 64Bs of data in FIFO below which the Pipe B Cursor stream will generate requests to memory |

### 2.8.3 WM1—Low Power 1 Display Watermarks

| WM1—Low Power 1 Display Watermarks | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **45108h-4510Bh** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | R/W |
| **Size (in bits):** | 32 |
| These watermark values will be used only when one pipe is enabled and no sprites are enabled (or the conditions for using the Low Power 1 Sprite Watermark are met) and the display is in LP1 state. | |

| Bit | Description |
|---|---|
| 31 | **Enabled** Project: All<br>Enables LP1 watermarks |
| 30:24 | **Latency** Project: All<br>The latency associated with the LP1 watermarks in half usecs. |
| 23:20 | **FBC_LP1_Watermark** Project: All<br>Number of equivalent lines of the primary display for this WM |
| 19:17 | **Reserved** Project: All Format: |
| 16:8 | **LP1_Primary_Watermark** Project: All<br>Number in 64Bs of data in FIFO below which the Primary stream will generate requests to memory. |
| 7:6 | **Reserved** Project: All Format: |
| 5:0 | **LP1_Cursor_Watermark** Project: All<br>Number in 64Bs of data in FIFO below which the Cursor stream will generate requests to memory. |

## 2.8.4 WM2—Low Power 2 Display Watermarks

### WM2—Low Power 2 Display Watermarks

**Register Type:** MMIO
**Address Offset:** **4510Ch-4510Fh**
**Project:** All
**Default Value:** 00000000h
**Access:** R/W
**Size (in bits):** 32

These watermark values will be used only when one pipe is enabled and no sprites are enabled and the display is in LP2 state.

| Bit | Description |
|---|---|
| 31 | **Enabled** Project: All<br>Enables LP2 watermarks<br>LP2 watermarks must be disabled when the Low Power 1 Sprite Watermark is enabled.<br>[DevILK] This bit must be programmed to 0 when FBC is enabled and FBC watermarks are disabled. |
| 30:24 | **Latency** Project: All<br>The latency associated with the LP2 watermarks in half usecs. |
| 23:20 | **FBC_LP2_Watermark** Project: All<br>Number of equivalent lines of the primary display for this WM |
| 19:17 | **Reserved** Project: All Format: |
| 16:8 | **LP2_Primary_Watermark** Project: All<br>Number in 64Bs of data in FIFO below which the Primary stream will generate requests to memory. |
| 7:6 | **Reserved** Project: All Format: |
| 5:0 | **LP2_Cursor_Watermark** Project: All<br>Number in 64Bs of data in FIFO below which the Cursor stream will generate requests to memory. |

## 2.8.5 WM3—Low Power 3 Display Watermarks

| WM3—Low Power 3 Display Watermarks | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **45110h-45113h** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | R/W |
| **Size (in bits):** | 32 |

These watermark values will be used only when one pipe is enabled and no sprites are enabled and the display is in LP3 state.

| Bit | Description |
|---|---|
| 31 | **Enabled** Project: All<br>Enables LP3 watermarks |
| 30:24 | **Latency** Project: All<br>The latency associated with the LP3 watermarks in half usecs. |
| 23:20 | **FBC_LP3_Watermark** Project: All<br>Number of equivalent lines of the primary display for this WM |
| 19:17 | **Reserved** Project: All Format: |
| 16:8 | **LP3_Primary_Watermark** Project: All<br>Number in 64Bs of data in FIFO below which the Primary stream will generate requests to memory. |
| 7:6 | **Reserved** Project: All Format: |
| 5:0 | **LP3_Cursor_Watermark** Project: All<br>Number in 64Bs of data in FIFO below which the Cursor stream will generate requests to memory. |

## 2.8.6 WM1S—Low Power 1 Sprite Watermark

| WM1S—Low Power 1 Sprite Watermark | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **45120h-45123h** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | R/W |
| **Size (in bits):** | 32 |

This watermark will be used only when one pipe is enabled and a sprite is enabled and sprite scaling is not enabled and the display is in LP1 state.

| Bit | Description |
|---|---|
| 31 | **Enabled** Project: All<br>Enables LP1 Sprite watermark. This bit allows memory self refresh to be entered when sprite is enabled. The WM2 Low Power 2 Display Watermark needs to be disabled when this watermark is enabled. This bit should only be changed when sprite is not enabled or display LP1 watermarks are not enabled. |
| 30:8 | **Reserved** Project: All Format: |
| 7:0 | **LP1_Sprite_Watermark**<br>Project: All<br>Default Value: 0b<br>Number in 64Bs of data in FIFO below which the Sprite stream will generate requests to memory. |

## 2.9 Backlight Control and Modulation Histogram Registers

### 2.9.1 BLC_PWM_CTL2—Backlight PWM Control Register 2

**BLC_PWM_CTL2—Backlight PWM Control Register 2**

**Register Type:** MMIO
**Address Offset:** **48250h-48253h**
**Project:** All
**Default Value:** 00000000h
**Access:** R/W
**Size (in bits):** 32

| Bit | Description |
|---|---|
| 31 | **PWM_Enable**<br>Project: All<br>Default Value: 0b<br>This bit enables the PWM counter logic<br>**Value** / **Name** / **Description** / **Project**<br>0b / Disable / PWM disabled (drives 0 always) / All<br>1b / Enable / PWM enabled / All |
| 30 | **Reserved** Project: All Format: |
| 29 | **PWM_Pipe_assignment**<br>Project: All<br>Default Value: 0b<br>This bit assigns PWM to a pipe. The PWM counter will run off of this pipe's PLL. The PWM function must be disabled in order to change the value of this field.<br>**Value** / **Name** / **Description** / **Project**<br>0b / Pipe A / Pipe A / All<br>1b / Pipe B / Pipe B / All |
| 28:27 | **Reserved** Project: All Format: |
| 26 | **Phase-In_Interrupt_Status**<br>Project: All<br>Access: R/W Clear<br>Default Value: 0b<br>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. |
| 25 | **Phase_In_Enable** Project: All<br>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. |

## BLC_PWM_CTL2—Backlight PWM Control Register 2

| | |
|---|---|
| 24 | **Phase_In_Interrupt_Enable** Project: All<br>Setting this bit enables an interrupt to be generated when the PWM phase in is completed. |
| 23:16 | **Phase_In_time_base**<br>Project: All<br>Default Value: 0b<br>This field determines the number of VBLANK events that pass before one increment occurs. |
| 15:8 | **Phase_In_Count** Project: All<br>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. |
| 7:0 | **Phase_In_Increment** Project: All<br>This field indicates the amount to adjust the PWM duty cycle register on each increment event.<br>This is a two's complement number. |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b [Default] | | Invalid | All |
| 01h-FFh | | VBlank Count | All |

## 2.9.2 BLC_PWM_CTL—Backlight PWM Control Register

## BLC_PWM_CTL—Backlight PWM Control Register

**Register Type:** MMIO
**Address Offset:** **48254h-48257h**
**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<br>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.<br>This register must be written only as a full 32 bit dword. Byte or word writes are not supported. |

### 2.9.3 BLM_HIST_CTL—Image Enhancement Histogram Control Register

## BLM_HIST_CTL—Image Enhancement Histogram Control Register

**Register Type:** MMIO
**Address Offset:** **48260h-48263h**
**Project:** All
**Default Value:** 00000000h
**Access:** R/W
**Size (in bits):** 32

| Bit | Description |
|---|---|
| 31 | **Image_Enhancement_Histogram_Enabled**<br>Project: All<br>Default Value: 0b<br>This bit enables the Image Enhancement histogram logic to collect data. |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Disable | Image histogram is disabled | All |
| 1b | Enable | 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. | All |

| Bit | Description |
|---|---|
| 30 | **Image_Enhancement_Modification_Table_Enabled**<br>Project: All<br>Default Value 0b<br>This bit enables the Image Enhancement modification table. |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Disable | Disabled | All |
| 1b | Enable | Enabled. When this bit is changed from a zero to a one, modifications begin after the next VBLANK of the assigned pipe. | All |

| Bit | Description |
|---|---|
| 29 | **Image_Enhancement_Pipe_assignment**<br>Project: All<br>Default Value: 0b<br>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. |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Pipe A | Pipe A | All |
| 1b | Pipe B | Pipe B | All |

| Bit | Description |
|---|---|
| 28:25 | **Reserved** Project: All Format: |

## BLM_HIST_CTL—Image Enhancement Histogram Control Register

**24 — Histogram Mode Select**

Project: All

Default Value: 0b

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | YUV | YUV Luma Mode | All |
| 1b | HSV | HSV Intensity Mode | All |

**23:16 — Sync_to_Phase_In_Count** 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.

**15 — Reserved** Project: All Format:

**14:13 — Enhancement_mode**

Project: All

Default Value: 00b

| Value | Name | Description | Project |
|---|---|---|---|
| 00b | Direct | Direct look up mode | All |
| 01b | Additive | Additive mode | All |
| 10b | Multiplicative | Multiplicative mode | All |
| 11b | Reserved | Reserved | All |

**12 — Sync_to_Phase_In** 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.

**11 — Bin_Register_Function_Select**

Project: All

Default Value: 0b

This field indicates what data is being written to or read from the bin data register.

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | BTC | Bin 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. | All |
| 1b | BIE | Bin Image Enhancement Value. Valid range for the Bin Index is 0 to 32 | All |

**10:7 — Reserved** Project: All Format:

**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 Image Enhancement Bin Data Register

### Image Enhancement Bin Data Register(F0 Threshold Count Usage)

| | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **48264h-48267h** |
| **Project:** | All |
| **Exists If:** | BLM_HIST_CTL:Bin Register Function Select = 0 |
| **Default Value:** | 00000000h |
| **Access:** | Read Only |
| **Size (in bits):** | 32 |

Writes to this address are steered to the correct register by programming the Bin Register Function Select and the Bin Register Index.

Function 0 usage (Threshold Count) this Function is Read Only

| Bit | Description |
|---|---|
| 31 | **Busy_Bit** Project: All<br>If set , the engine is busy, the rest of the register is undefined. If clear, the register contains valid data. |
| 30:22 | **Reserved** Project: All Format: |
| 21:0 | **Bin_Count** Project: All<br>The total number of pixels in this bin, value is updated at the start of each vblank. |

### Image Enhancement Bin Data Register(F1 Image Enhancement Usage)

| | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **48264h-48267h** |
| **Project:** | All |
| **Exists If:** | BLM_HIST_CTL:Bin Register Function Select = 1 |
| **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 |

Writes to this address are steered to the correct register by programming the Bin Register Function Select and the Bin Register Index.

Function 1 usage (Image Enhancement) this Function is Read/Write

| Bit | Description |
|---|---|
| 31:10 | **Reserved** Project: All Format: |
| 9:0 | **Image_Bin_Correction_Factor** Project: All<br>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 Histogram Threshold Guardband Register

### Histogram Threshold Guardband Register

**Register Type:** MMIO
**Address Offset:** **48268h-4826Bh**
**Project:** All
**Default Value:** 00000000h
**Access:** R/W
**Size (in bits):** 32
**Double Buffer Update Point:** Start of vertical blank

| Bit | Description |
|---|---|
| 31 | **Histogram_Interrupt_enable**<br>Project: All<br>Default Value: 0b |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Disable | Disabled | All |
| 1b | Enable | This generates a histogram interrupt once a Histogram event occurs. Software must always program 1. | All |

| Bit | Description |
|---|---|
| 30 | **Histogram_Event_status**<br>Project: All<br>Access: R/W Clear<br>Default Value: 0b<br>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' |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Not Occurred | Histogram event has not occurred | All |
| 1b | Occurred | Histogram event has occurred | All |

| Bit | Description |
|---|---|
| 29:22 | **Guardband_Interrupt_Delay** Project: All<br>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. |
| 21:0 | **Threshold_Guardband** Project: All<br>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 |

## 2.10 Motion Blur Mitigation (MBM) Control

These registers are use to control the MBM logic. (Sometimes called LOT, panel overdrive, or LRTC).

Before enabling MBM, software should have identically programmed both pipes source size and gamma tables. 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 second plane(s), this can be done by MMIO or by a flip command. The second 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

**MBM_CTRL—MBM Control**

**Register Type:** MMIO
**Address Offset:** **48800h-48803h**
**Project:** All
**Default Value:** 00000000h
**Access:** R/W
**Size (in bits):** 32

| Bit | Description |
|---|---|
| 31 | **MBM_Enable**<br>Project: All<br>Default Value: 0b<br>This bit enables MBM logic. |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Disable | MBM is Disabled | All |
| 1b | Enable | MBM is Enabled | All |

| Bit | Description |
|---|---|
| 30 | **Reserved** Project: All Format: |
| 29 | **MBM Pipe Select**<br>Project: All<br>Default Value: 0b<br>This bit assigns MBM modification to the selected pipe. |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Pipe A | PipeA (PipeA will fetch the current buffer and PipeB will fetch the previous buffer) | All |
| 1b | Pipe B | PipeB (PipeB will fetch the current buffer and PipeA will fetch the previous buffer) | All |

| MBM_CTRL—MBM Control | |
|---|---|
| 28:27 | **MBM_Surface_select**<br>Project: All<br>Default Value: 0b<br><br>**Value** / **Name** / **Description** / **Project**<br>00b / None / None / All<br>01b / Sprite / Sprite Only / All<br>10b / Primary / Primary Only / All<br>11b / Both / Both sprite and primary / All |
| 26:24 | **Reserved** Project: All Format: |
| 23:16 | **MBM_Delta_Threshold**<br>Project: All<br>Default Value: 00000000b<br>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. |
| 15:0 | **Reserved** Project: All Format: |

## 2.10.2 MBM_TBL—MBM Overdrive Table

| MBM_TBL—MBM Overdrive Table | | | |
|---|---|---|---|
| **Register Type:** MMIO<br>**Address Offset:** **48810h-4890Bh**<br>**Project:** All<br>**Default Value:** 00000000h<br>**Access:** R/W<br>**Size (in bits):** 63x32 | | | |
| These will be the overdrive values to be 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. | | | |
| **DWord** | **Bit** | **Description** | |
| 0..62 | 31:24 | **Reserved** Project: All Format: | |
| | 23:16 | **Red_MBM_overdrive_value** | Project: All |
| | 15:8 | **Green_MBM_overdrive_value** | Project: All |
| | 7:0 | **Blue_MBM_overdrive_value** | Project: All |

**Overdrive table address offsets:**

| | 0 | 31 | 63 | 95 | 127 | 159 | 191 | 223 | 255 |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **31** | 48810h | 48814h | 48818h | 4881Ch | 48820h | 48824h | 48828h | 4882Ch | 48830h |
| **63** | 48834h | 48838h | 4883Ch | 48840h | 48844h | 48848h | 4884Ch | 48850h | 48854h |
| **95** | 48858h | 4885Ch | 48860h | 48864h | 48868h | 4886Ch | 48870h | 48874h | 48878h |
| **127** | 4887Ch | 48880h | 48884h | 48888h | 4888Ch | 48890h | 48894h | 48898h | 4889Ch |
| **159** | 488A0h | 488A4h | 488A8h | 488ACh | 488B0h | 488B4h | 488B8h | 488BCh | 488C0h |
| **191** | 488C4h | 488C8h | 488CCh | 488D0h | 488D4h | 488D8h | 488DCh | 488E0h | 488E4h |
| **223** | 488E8h | 488ECh | 488F0h | 488F4h | 488F8h | 488FCh | 48900h | 48904h | 48908h |
| **255** | 256 | 256 | 256 | 256 | 256 | 256 | 256 | 256 | 256 |

Previous Pixel Value Range

Current Pixel Value Range

Address Offset

Hard Coded Value

## 2.11 Color Conversion and Control Registers

These registers contain the coefficients of the pipe color space converter. There are 12 values in 6 registers for each pipe. This color space conversion is used to convert the RGB frame buffer data into YUV data for use on the HDMI or DisplayPort. Or alternately a YUV frame buffer could be converted to RGB.

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.

## CSC Coefficient Description

**Project:** All
**Default Value:** 0000h
**Size (in bits):** 16

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.

| Bit | Description |
|---|---|
| 15 | **Sign**<br>Project: All<br><br>**Value** / **Name** / **Description** / **Project**<br>0b / Positive / Positive / All<br>1b / Negative / Negative / All |
| 14:12 | **Exponent_bits**<br>Project: All<br>Represented as $2^{-n}$<br><br>**Value** / **Name** / **Description** / **Project**<br>110b / 4 / 4 or mantissa is bb.bbbbbbb / All<br>111b / 2 / 2 or mantissa is b.bbbbbbbb / All<br>000b / 1 / 1 or mantissa is 0.bbbbbbbbb / All<br>001b / 0.5 / 0.5 or mantissa is 0.0bbbbbbbbb / All<br>010b / 0.25 / 0.25 or mantissa is 0.00bbbbbbbbb / All<br>011b / 0.125 / 0.125 or mantissa is 0.000bbbbbbbbb / All<br>others / Reserved / Reserved / All |
| 11:3 | **Mantissa** Project: All |
| 2:0 | **Reserved** Project: All Format: |

The matrix equations are as follows:

Y = (RY * R) + (GY * G) + (BY * B)

U = (RU * R) + (GU * G) + (BU * B)

V = (RV * R) + (GV * G) + (BV * B)

The standard programming for RGB to YUV is in the following table.

| | 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 |

The standard programming for YUV to sRGB without scaling 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 the pre-CSC offsets to -128, -16, and -128 for high, medium, and low channels respectively.

The coefficients 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 |

## 2.11.1 Pipe A Color Control

### 2.11.1.1 CSC_A_Coefficients 1

| CSC_A_Coefficients 1 | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **49010h-49013h** |
| **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 CSC_A_Mode |

| Bit | Description |
|---|---|
| 31:16 | **RY**<br>Project: All<br>Format: CSC COEFFICIENT DESCRIPTION<br>FormatDesc: 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.<br>CSC coefficient. See format description above. |
| 15:0 | **GY**<br>Project: All<br>Format: CSC COEFFICIENT DESCRIPTION<br>FormatDesc: 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.<br>CSC coefficient. See format description above. |

### 2.11.1.2 CSC_A_Coefficients 2

| CSC_A_Coefficients 2 | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **49014h-49017h** |
| **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 CSC_A_Mode |

| Bit | Description |
|---|---|
| 31:16 | **BY**<br>Project: All<br>Format: CSC COEFFICIENT DESCRIPTION<br>FormatDesc: 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.<br>CSC coefficient. See format description above. |
| 15:0 | **Reserved** Project: All Format: |

### 2.11.1.3 CSC_A_Coefficients 3

| CSC_A_Coefficients 3 | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **49018h-4901Bh** |
| **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 CSC_A_Mode |

| Bit | Description |
|---|---|
| 31:16 | **RU**<br>Project: All<br>Format: CSC COEFFICIENT DESCRIPTION<br>FormatDesc: 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.<br>CSC coefficient. See format description above. |
| 15:0 | **GU**<br>Project: All<br>Format: CSC COEFFICIENT DESCRIPTION<br>FormatDesc: 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.<br>CSC coefficient. See format description above. |

### 2.11.1.4 CSC_A_Coefficients 4

| CSC_A_Coefficients 4 | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **4901Ch-4901Fh** |
| **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 CSC_A_Mode |

| Bit | Description |
|---|---|
| 31:16 | **BU**<br>Project: All<br>Format: CSC COEFFICIENT DESCRIPTION<br>FormatDesc: 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.<br>CSC coefficient. See format description above. |
| 15:0 | **Reserved** Project: All Format: |

### 2.11.1.5 CSC_A_Coefficients 5

| CSC_A_Coefficients 5 | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **49020h-49023h** |
| **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 CSC_A_Mode |

| Bit | Description |
|---|---|
| 31:16 | **RV**<br>Project: All<br>Format: CSC COEFFICIENT DESCRIPTION<br>FormatDesc: 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.<br>CSC coefficient. See format description above. |
| 15:0 | **GV**<br>Project: All<br>Format: CSC COEFFICIENT DESCRIPTION<br>FormatDesc: 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.<br>CSC coefficient. See format description above. |

### 2.11.1.6 CSC_A_Coefficients 6

| CSC_A_Coefficients 6 | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **49024h-49027h** |
| **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 CSC_A_Mode |

| Bit | Description |
|---|---|
| 31:16 | **BV**<br>Project: All<br>Format: CSC COEFFICIENT DESCRIPTION<br>FormatDesc: 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.<br>CSC coefficient. See format description above. |
| 15:0 | **Reserved** Project: All Format: |

### 2.11.1.7 CSC_A_Mode

## CSC_A_Mode

| | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **49028h-4902Bh** |
| **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_A registers**

| Bit | Description |
|---|---|
| 31:3 | **Reserved** Project: All Format: |
| 2 | **CSC_Black_Screen_Offset**<br>Project: All<br>Default Value: 0b<br>Adds an offset to the data output from CSC<br>In sRGB ouput mode: RGB is defined as R`+ 1/16, G`+ 1/16, B`+ 1/16<br>In rcYUV output mode: YUV is defined as Y'+ 1/16, U and V are output in excess 2048 format |
| 1 | **CSC_Position**<br>Project: All<br>Default Value: 0b<br>Selects the CSC position in the pipe. |
| 0 | **CSC_Mode**<br>Project: All<br>Default Value: 0b<br>Selects the CSC direction. |

**Bit 2**

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | No Offset | CSC output has no offset added (will be RGB or YUV, depending on bit 0) | All |
| 1b | Offset | CSC output has offset added (will be sRGB or rcYUV depending on bit 0) | All |

**Bit 1**

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | CSC After | CSC is after gamma and DPST image enhancement | All |
| 1b | CSC Before | CSC is before gamma and DPST image enhancement | All |

**Bit 0**

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | RGB to YUV | RGB to YUV conversion | All |
| 1b | YUV to RGB | YUV to RGB conversion | All |

### 2.11.1.8 Pre-CSC_A High Color Channel Offset

| Pre-CSC_A High Color Channel Offset | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **49030h-49033h** |
| **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 CSC_A_Mode |

| Bit | Description |
|---|---|
| 31:13 | **Reserved** Project: All Format: |
| 12:0 | **Pre-CSC_High_Color_Channel_Offset** Project: All<br>This 13-bit 2's complement value is used to give an offset to the color channel as it enters CSC logic. |

### 2.11.1.9 Pre-CSC_A Medium Color Channel Offset

| Pre-CSC_A Medium Color Channel Offset | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **49034h-49037h** |
| **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 CSC_A_Mode |

| Bit | Description |
|---|---|
| 31:13 | **Reserved** Project: All Format: |
| 12:0 | **Pre-CSC_Medium_Color_Channel_Offset** Project: All<br>This 13-bit 2's complement value is used to give an offset to the color channel as it enters CSC logic. |

#### 2.11.1.10 Pre-CSC_A Low Color Channel Offset

| Pre-CSC_A Low Color Channel Offset | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **49038h-4903Bh** |
| **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 CSC_A_Mode |

| Bit | Description |
|---|---|
| 31:13 | **Reserved** Project: All Format: |
| 12:0 | **Pre-CSC_Low_Color_Channel_Offset** Project: All<br>This 13-bit 2's complement value is used to give an offset to the color channel as it enters CSC logic. |

### 2.11.2 Pipe B Color Control

#### 2.11.2.1 CSC_B_Coefficients 1

| CSC_B_Coefficients 1 | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **49110h-49113h** |
| **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 CSC_B_Mode |

| Bit | Description |
|---|---|
| 31:16 | **RY**<br>Project: All<br>Format: CSC COEFFICIENT DESCRIPTION<br>FormatDesc: 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.<br>CSC coefficient. See format description above. |

| CSC_B_Coefficients 1 | |
|---|---|
| 15:0 | **GY**<br>Project: All<br>Format: CSC COEFFICIENT DESCRIPTION<br>FormatDesc: 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.<br>CSC coefficient. See format description above. |

### 2.11.2.2 CSC_B_Coefficients 2

| CSC_B_Coefficients 2 | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **49114h-49117h** |
| **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 CSC_B_Mode |

| Bit | Description |
|---|---|
| 31:16 | **BY**<br>Project: All<br>Format: CSC COEFFICIENT DESCRIPTION<br>FormatDesc: 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.<br>CSC coefficient. See format description above. |
| 15:0 | **Reserved** Project: All Format: |

### 2.11.2.3 CSC_B_Coefficients 3

| CSC_B_Coefficients 3 | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **49118h-4911Bh** |
| **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 CSC_B_Mode |

| Bit | Description |
|---|---|
| 31:16 | **RU**<br>Project: All<br>Format: CSC COEFFICIENT DESCRIPTION<br>FormatDesc: 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.<br>CSC coefficient. See format description above. |
| 15:0 | **GU**<br>Project: All<br>Format: CSC COEFFICIENT DESCRIPTION<br>FormatDesc: 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.<br>CSC coefficient. See format description above. |

### 2.11.2.4 CSC_B_Coefficients 4

| CSC_B_Coefficients 4 | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **4911Ch-4911Fh** |
| **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 CSC_B_Mode |

| Bit | Description |
|---|---|
| 31:16 | **BU**<br>Project: All<br>Format: CSC COEFFICIENT DESCRIPTION<br>FormatDesc: 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.<br>CSC coefficient. See format description above. |
| 15:0 | **Reserved** Project: All Format: |

### 2.11.2.5 CSC_B_Coefficients 5

| CSC_B_Coefficients 5 | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **49120h-49123h** |
| **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 CSC_B_Mode |

| Bit | Description |
|---|---|
| 31:16 | **RV**<br>Project: All<br>Format: CSC COEFFICIENT DESCRIPTION<br>FormatDesc: 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.<br>CSC coefficient. See format description above. |
| 15:0 | **GV**<br>Project: All<br>Format: CSC COEFFICIENT DESCRIPTION<br>FormatDesc: 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.<br>CSC coefficient. See format description above. |

### 2.11.2.6 CSC_B_Coefficients 6

| CSC_B_Coefficients 6 | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **49124h-49127h** |
| **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 CSC_B_Mode |

| Bit | Description |
|---|---|
| 31:16 | **BV**<br>Project: All<br>Format: CSC COEFFICIENT DESCRIPTION<br>FormatDesc: 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.<br>CSC coefficient. See format description above. |
| 15:0 | **Reserved** Project: All Format: |

### 2.11.2.7 CSC_B_Mode

| CSC_B_Mode | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **49128h-4912Bh** |
| **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_B registers**

| Bit | Description |
|---|---|
| 31:3 | **Reserved** Project: All Format: |

## CSC_B_Mode

2

**CSC_Black_Screen_Offset**

Project: All

Default Value: 0b

Adds an offset to the data output from CSC

In sRGB ouput mode: RGB is defined as R`+ 1/16, G`+ 1/16, B`+ 1/16

In rcYUV output mode: YUV is defined as Y'+ 1/16, U and V are output in excess 2048 format

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | No Offset | CSC output has no offset added (will be RGB or YUV, depending on bit 0) | All |
| 1b | Offset | CSC output has offset added (will be sRGB or rcYUV depending on bit 0) | All |

1

**CSC_Position**

Project: All

Default Value: 0b

Selects the CSC position in the pipe.

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | CSC After | CSC is after gamma and DPST image enhancement | All |
| 1b | CSC Before | CSC is before gamma and DPST image enhancement | All |

0

**CSC_Mode**

Project: All

Default Value: 0b

Selects the CSC direction. Input and output formats and position within the pipe

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | RGB to YUV | RGB to YUV conversion | All |
| 1b | YUV to RGB | YUV to RGB conversion | All |

### 2.11.2.8 Pre-CSC_B High Color Channel Offset

| Pre-CSC_B High Color Channel Offset | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **49130h-49133h** |
| **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 CSC_B_Mode |

| Bit | Description |
|---|---|
| 31:13 | **Reserved** Project: All Format: |
| 12:0 | **Pre-CSC_B_High_Color_Channel_Offset** Project: All<br>This 13-bit 2's complement value is used to give an offset to the color channel as it enters CSC logic. |

### 2.11.2.9 Pre-CSC_B Medium Color Channel Offset

| Pre-CSC_B Medium Color Channel Offset | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **49134h-49137h** |
| **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 CSC_B_Mode |

| Bit | Description |
|---|---|
| 31:13 | **Reserved** Project: All Format: |
| 12:0 | **Pre-CSC_B_Medium_Color_Channel_Offset** Project: All<br>This 13-bit 2's complement value is used to give an offset to the color channel as it enters CSC logic. |

### 2.11.2.10 Pre-CSC_B Low Color Channel Offset

| Pre-CSC_B Low Color Channel Offset | | |
|---|---|---|
| **Register Type:** | MMIO | |
| **Address Offset:** | **49138h-4913Bh** | |
| **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 CSC_B_Mode | |
| **Bit** | **Description** | |
| 31:13 | **Reserved** Project: All Format: | |
| 12:0 | **Pre-CSC_B_Low_Color_Channel_Offset** Project: All<br>This 13-bit 2's complement value is used to give an offset to the color channel as it enters CSC logic. | |

## 2.12 Display Palette Registers (4A000h–4CFFFh)

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 three different modes.

8 bit legacy palette mode (for indexed VGA and 8 bpp formats and for legacy gamma):

256 entries of 24 bits each (8 bits per color).

For indexed formats, an 8 bit per pixel value is used to lookup a 24 bit per pixel value from the palette which then is padded to 36 bits. This permits a compact data format to choose from 256 colors out of a larger palette of colors. The legacy palette is accessible through both MMIO and VGA palette register I/O addresses. Through VGA palette register I/O addresses, the palette can look as though there are only 6 bits per color component (this mapping is handled inside the VGA engine).

For legacy gamma, the 36 bits per pixel gamma input is chopped to 24 bits and used to lookup a 24 bit pixel value from the palette which then is padded to 36 bits. This permits a color to be re-mapped to a different brightness for gamma correction. This provides the lowest quality gamma and should only be used for legacy requirements.

10 bit precision palette mode (for 10 bit gamma):

1024 entries of 30 bits each (10 bits per color).

For 10 bit gamma, the 36 bits per pixel gamma input is chopped to 30 bits and used to lookup a 30 bit pixel value from the palette which then is padded to 36 bits. This permits a color to be re-mapped to a different brightness for gamma correction. This provides the highest quality gamma for non-indexed pixel data formats of 30 bits per pixel or less.

12 bit interpolated gamma mode:

512 entries of 16 bits each (format described in 12 bit interpolated gamma programming notes).

For 12 bit interpolated gamma, the 36 bits per pixel gamma input is used to lookup reference points (16 bits per color in 12.4 format) along a gamma curve and interpolate a 36 bit pixel result. This permits a color to be re-mapped to a different brightness for gamma correction. This provides the highest quality gamma for pixel data formats greater than 30 bits per pixel.

Pixel chopping refers to removing the LSBs of each color component to reduce bits per pixel. Pixel padding refers to adding LSBs to each color component to increase bits per pixel.

Accesses to the palette entries require that the core display clock is running at the time of the update. Do not access the palette if the pipe is enabled and mode set is not yet complete. All write accesses to the palette must be in dwords. Byte or word writes to the palettes are not allowed.

## 2.12.1 LGC_PALETTE_A—Pipe A Legacy Display Palette

| LGC_PALETTE_A—Pipe A Legacy Display Palette | | |
|---|---|---|
| **Register Type:** | MMIO | |
| **Address Offset:** | **4A000h-4A3FFh** | |
| **Project:** | All | |
| **Default Value:** | UUUUUUUUh | |
| **Access:** | R/W (DWORD access only, no byte access) | |
| **Size (in bits):** | 256x32 | |
| **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 LGC_PALETTE_B—Pipe B Legacy Display Palette

| LGC_PALETTE_B—Pipe B Legacy Display Palette | | |
|---|---|---|
| **Register Type:** | MMIO | |
| **Address Offset:** | **4A800h-4ABFFh** | |
| **Project:** | All | |
| **Default Value:** | UUUUUUUUh | |
| **Access:** | R/W (DWORD access only, no byte access) | |
| **Size (in bits):** | 256x32 | |
| **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: |

## 2.12.3 PREC_PALETTE_A—Pipe A Precision Display Palette

| 10 bit Precision Palette Mode Format | | |
|---|---|---|
| **Project:** | All | |
| Format for 10 bit precision palette mode. | | |
| **Bit** | **Description** | |
| 31:30 | **Reserved** Project: All | Format: |
| 29:20 | **Red_Palette_Entry** | Project: All |
| 19:10 | **Green_Palette_Entry** | Project: All |
| 9:0 | **Blue_Palette_Entry** | Project: All |

| 12-bit Interpolated Precision Palette Mode (odd Dword) Format | | |
|---|---|---|
| **Project:** | All | |
| Format for 12 bit interpolated gamma mode, odd dwords. | | |
| **Bit** | **Description** | |
| 31:30 | **Reserved** Project: All | Format: |
| 29:20 | **Red_ Base[11:2]** | Project: All |
| 19:10 | **Green_ Base[11:2]** | Project: All |
| 9:0 | **Blue_ Base[11:2]** | Project: All |

## 12-bit Interpolated Precision Palette Mode (even Dword) Format

**Project:** All

Format for 12 bit interpolated gamma mode, odd dwords..

| Bit | Description |
|---|---|
| 31:30 | **Reserved** Project: All Format: MBZ |
| 29:28 | **Red_Base[1:0]** Project: All |
| 27:24 | **Red_Fraction** Project: All |
| 23:20 | **Reserved** Project: All Format: |
| 19:18 | **Green_Base[1:0]** Project: All |
| 17:14 | **Green_Fraction** Project: All |
| 13:10 | **Reserved** Project: All Format: |
| 9:8 | **Blue_Base[1:0]** Project: All |
| 7:4 | **Blue_Fraction** Project: All |
| 3:0 | **Reserved** Project: All Format: |

## PREC_PALETTE_A—Pipe A Precision Display Palette(10 bit)

**Register Type:** MMIO
**Address Offset:** **4B000h-4BFFFh**
**Project:** All
**Exists If:** PIPEACONF:Pipe_A_Palette/Gamma_Unit_mode = 01b
**Default Value:** UUUUUUUUh
**Access:** R/W (DWORD access only, no byte access)
**Size (in bits):** 1024x32

| DWord | Bit | Description |
|---|---|---|
| 0..1023 | 31:0 | **10bit_mode** Project: All Format: 10 bit Precision Palette Mode Format<br><br>See format description above |

| PREC_PALETTE_A—Pipe A Precision Display Palette(12 bit) | | |
|---|---|---|
| **Register Type:** | MMIO | |
| **Address Offset:** | **4B000h-4BFFFh** | |
| **Project:** | All | |
| **Exists If:** | PIPEACONF:Pipe_A_Palette/Gamma_Unit_mode = 10b | |
| **Default Value:** | UUUUUUUUh | |
| **Access:** | R/W (DWORD access only, no byte access) | |
| **Size (in bits):** | 1024x32 | |
| **DWord** | **Bit** | **Description** |
| 0..1023 | 63:32 | **12bit_odd** Project: All Format: 12-bit Interpolated Precision Palette Mode (odd Dword) Format<br>FormatDesc: Format for 12 bit interpolated gamma mode, odd dwords. |
| | 31:0 | **12bit_even** Project: All Format: 12-bit Interpolated Precision Palette Mode (even Dword) Format<br>FormatDesc: Format for 12 bit interpolated gamma mode dwords |

## 2.12.4 PREC_PALETTE_B—Pipe B Precision Display Palette

| PREC_PALETTE_B—Pipe B Precision Display Palette(10 bit) | | |
|---|---|---|
| **Register Type:** | MMIO | |
| **Address Offset:** | **4C000h-4CFFFh** | |
| **Project:** | All | |
| **Exists If:** | PIPEBCONF:Pipe_B_Palette/Gamma_Unit_mode = 01b | |
| **Default Value:** | UUUUUUUUh | |
| **Access:** | R/W (DWORD access only, no byte access) | |
| **Size (in bits):** | 1024x32 | |
| **DWord** | **Bit** | **Description** |
| 0..1023 | 31:0 | **10bit_mode** Project: All Format: 10 bit Precision Palette Mode Format<br>FormatDesc: Format for 10 bit precision palette mode. |

## PREC_PALETTE_B—Pipe B Precision Display Palette(12 bit)

| | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **4C000h-4CFFFh** |
| **Project:** | All |
| **Exists If:** | PIPEBCONF:Pipe_B_Palette/Gamma_Unit_mode = 10b |
| **Default Value:** | UUUUUUUUh |
| **Access:** | R/W (DWORD access only, no byte access) |
| **Size (in bits):** | 1024x32 |

| DWord | Bit | Description |
|---|---|---|
| 0..1023 | 63:32 | **12bit_odd** Project: All Format: 12-bit Interpolated Precision Palette Mode (odd Dword) Format<br>Format Desc: Format for 12 bit interpolated gamma mode, odd dwords. |
| | 31:0 | **12bit_even** Project: All Format: 12-bit Interpolated Precision Palette Mode (even Dword) Format<br>FormatDesc: Format for 12 bit interpolated gamma mode dwords. |

**12-bit Interpolated Gamma Programming Notes:**

The 12-bit gamma correction curve is represented by specifying a set of reference points spaced equally along the desired curve. Red, Green, and Blue each have 513 reference points. The first 512 reference points are stored in the precision palette RAM, and the final value is stored in the GCMAX register. The first 512 reference points are 16 bits represented in a 12.4 format with 12 integer and 4 fractional bits. The final reference points are 17 bits represented in a 13.4 format with 13 integer and 4 fractional bits.

During operation the appropriate reference point pairs (adjacent) are selected for each color, and the output is interpolated between these two reference point values.

To program the gamma correction reference points, calculate the desired gamma curve for inputs from 0 to 4096. Every 8th point on the curve (0, 8, 16 ... 4088, 4096) becomes a reference point. Convert the gamma value to the 13.4 format. The first 512 reference points are saved to the precision palette RAM, where the even DWords contain the lower 6 bits of the reference point value, and the odd DWords contain the upper 10 bits of the reference point value. The final 513th reference point is saved in the GCMAX registers in 13.4 format.

Example equation for gamma curve of 2.2:

For (X = 0..4096) { gamma = [(X / 4096) ^ 2.2] * 4096 }

The curve must be flat or increasing, never decreasing.

## 2.12.5 PIPEAGCMAX—Pipe A Gamma Correction Max

| Pipe Max Gamma Correction Format | | |
|---|---|---|
| **Project:** | All | |
| **Default Value:** | 00010000h | |
| **Bit** | **Description** | |
| 31:17 | **Reserved** Project: All | Format: |
| 16:0 | **Max_Color_Gamma_Correction_Point** 513th reference point for the color channel of the 12-bit pipe piecewise linear gamma correction. The value should always be programmed to be less than or equal to 4096.0. | Project: All |

| PIPEAGCMAX—Pipe A Gamma Correction Max | | | | | | |
|---|---|---|---|---|---|---|
| **Register Type:** | MMIO | | | | | |
| **Address Offset:** | **4D000h-4D00Bh** | | | | | |
| **Project:** | All | | | | | |
| **Default Value:** | 00010000h | | | | | |
| **Access:** | R/W | | | | | |
| **Size (in bits):** | 3x32 | | | | | |
| **DWord** | **Bit** | **Description** | | | | |
| 0 | 31:0 | **Red** | Project: | All | Format: | Pipe Max Gamma Correction Format |
| 1 | 31:0 | **Green** | Project: | All | Format: | Pipe Max Gamma Correction Format |
| 2 | 31:0 | **Blue** | Project: | All | Format: | Pipe Max Gamma Correction Format |

### 2.12.6 PIPEBGCMAX —Pipe B Gamma Correction Max

| PIPEBGCMAX—Pipe B Gamma Correction Max | | | | | | |
|---|---|---|---|---|---|---|
| **Register Type:** MMIO<br>**Address Offset:** **4D010h-4D01Bh**<br>**Project:** All<br>**Default Value:** 00010000h<br>**Access:** R/W<br>**Size (in bits):** 3x32 | | | | | | |
| **DWord** | **Bit** | **Description** | | | | |
| 0 | 31:0 | **Red** | Project: | All | Format: | Pipe Max Gamma Correction Format |
| 1 | 31:0 | **Green** | Project: | All | Format: | Pipe Max Gamma Correction Format |
| 2 | 31:0 | **Blue** | Project: | All | Format: | Pipe Max Gamma Correction Format |

## 2.13 Software Flag Registers (4F000h–4F10Fh)

### 2.13.1 Software Flag Registers

| Software Flag Registers | | | | | | |
|---|---|---|---|---|---|---|
| **Register Type:** MMIO<br>**Address Offset:** **4F000h-4F08Fh**<br>**Project:** All<br>**Default Value:** 00000000h<br>**Access:** R/W<br>**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 | **Reserved** | Project: | All | Format: | PBC |

## 2.13.2 DE_LOAD_SL — Display Load Scan Lines

| DE_LOAD_SL—Display Load Scan Lines | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **4F100h-4F103h** |
| **Project:** | **DevSNB** |
| **Default Value:** | 00000000h |
| **Access:** | R/W |
| **Size (in bits):** | 32 |

This register is used to initiate a display scan line compare on DevSNB:D2 and later steppings.

When this register is written with the Init_Compare bit set to 1b, the Display Engine (DE) will start comparing the display pipe timing generator current scan line value (current scan line) with the start scan line value (current scan line >= start scan line) and the end scan line value (current scan line <= end scan line) to decide if the the pipe scan line is inside or outside the scan line window of interest. DE will wait until the current scan line is either outside (Inclusive mode) or inside (Exclusive mode) the scan line window, then trigger a scan line event and stop any further comparing.

The scan line event can cause display to send a scan line compare response to the render command streamer, used for releasing a MI_WAIT_FOR_EVENT on scan line window, if unmasked in the DERRMR mask register 0x44050. The scan line event can also cause display to generate a scan line compare interrupt, if unmasked and enabled in the DEIMR and DEIER display engine interrupt registers.

The end scan line value must be greater than or equal to the start scan line value.

The current scan line is a 13 bit value, but the scan line comparison is only done on the upper 10 bits. This gives an 8 line granularity on the compare.

The programmable range can include the vertical blank.

In interlaced display timings, the current scan line is the the current line of the current interlaced field.

Notes for command streamer programming to use this display load scan lines register:

- Either MMIO or a MI_LOAD_REGISTER_IMM command can be used to unmask the scan line render response 0x44050. That can be done any time before programming this register.
- In order to use MI_WAIT_FOR_EVENT on scan line window, DE_LOAD_SL must be programmed using MI_LOAD_REGISTER_IMM immediately prior to the MI_WAIT_FOR_EVENT on scan line window, both commands must be in the same cacheline, both commands must be executed using the same tail or batch update, and if sync flush is enabled, MI_SUSPEND_FLUSH must be used to suspend flushes prior to the commands.

## DE_LOAD_SL—Display Load Scan Lines

| Bit | Description |
|---|---|
| 31 | **Initiate_Compare**<br>Default Value: 0b<br>This field initiates the scan line compare.<br>When this register is written with this bit set to 1b, the display engine will do one complete comparison cycle, trigger a scan line event, then stop comparing.<br>Do not write this register again until after any previous scan line compare has completed.<br>A compare can not be cancelled by writing with this bit set to 0b. |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Do nothing | Do nothing | All |
| 1b | Initiate | Initiate the scan line compare. | All |

| Bit | Description |
|---|---|
| 30 | **Inclusive_Exclusive_Select**<br>Default Value: 0b<br>This field selects whether the scan line compare is done in exclusive mode, where display triggers the scan line event when inside the window (wait while outside window), or inclusive mode, where display triggers the scan line event when outside the scan line window (wait while inside window). |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Exclusive | Exclusive mode: trigger scan line event when inside the scan line window (wait while outside window) | All |
| 1b | Inclusive | Inclusive mode: trigger scan line event when outside the scan line window (wait while inside window) | All |

| Bit | Description |
|---|---|
| 29 | **Pipe_Select**<br>Default Value: 0b<br>This field selects which display pipe timing generator scan line value to use for the compare. |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Pipe A | Display Pipe A | All |
| 1b | Pipe B | Display Pipe B | All |

| Bit | Description |
|---|---|
| 28:19 | **Start_Scan_Line** Project: DevSNB Format:<br>This field specifies the upper 10 bits of the starting scan line number of the scan line window.<br>The lower 3 bits are reserved below and not programmable and not compared. |
| 18:13 | **Reserved** Project: All Format: |
| 12:3 | **End_Scan_Line** Project: DevSNB Format:<br>This field specifies the upper 10 bits of the ending scan line number of the scan line window.<br>The lower 3 bits are reserved below and not programmable and not compared. |
| 2:0 | **Reserved** Project: All Format: |

# 3. North Pipe and Port Controls (60000h–6FFFFh)

## 3.1 Pipe A Timing

### 3.1.1 HTOTAL_A—Pipe A Horizontal Total Register

| HTOTAL_A—Pipe A Horizontal Total Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **60000h-60003h** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | R/W |
| **Size (in bits):** | 32 |

| Bit | Description |
|---|---|
| 31:29 | **Reserved** Project: All Format: MBZ |
| 28:16 | **Pipe_A_Horizontal_Total_Display_Clocks** Project: All<br>This 13-bit field provides Horizontal Total up to 8192 pixels encompassing the Horizontal Active Display period, front/back border and retrace period. This field is programmed to the number of clocks desired minus one.<br>This number of clocks needs to be a multiple of two when driving the LVDS port in two channel mode. This value should always be equal or greater to the sum of the horizontal active and the horizontal blank, and border region sizes. |
| 15:12 | **Reserved** Project: All Format: MBZ |
| 11:0 | **Pipe_A_Horizontal_Active_Display_Pixels** Project: All<br>This 12-bit field provides Horizontal Active Display resolutions up to 4096 pixels. Note that the first horizontal active display pixel is considered pixel number 0. The value programmed should be the (active pixels/line – 1).<br>The number of active pixels will be limited to multiples of two pixels when driving the integrated LVDS port in two channel mode. The minimum horizontal active display size allowed will be 64 pixels. |

### 3.1.2 HBLANK_A—Pipe A Horizontal Blank Register

| HBLANK_A—Pipe A Horizontal Blank Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **60004h-60007h** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | R/W |
| **Size (in bits):** | 32 |

| Bit | Description |
|---|---|
| 31:29 | **Reserved** Project: All Format: |
| 28:16 | **Pipe_A_Horizontal_Blank_End** Project: All<br>This 13-bit field specifies the position of Horizontal Blank End expressed in terms of the absolute pixel number relative to the horizontal active display start. The value programmed should be the HBLANK End pixel position, where the first active pixel is considered position 0; the second active pixel is considered position 1, etc. Horizontal blank ending at the same point as the horizontal total indicates that there is no left hand border area. HBLANK size has a minimum value of 32 clocks.<br>The number of clocks within blank needs to be a multiple of two when driving the LVDS port in two channel mode.<br>The value loaded in the register would be equal to RightBorder+Active+HBlank-1.<br>The border must be 0, so this register must always be programmed to the same value as the Horizontal Total. |
| 15:13 | **Reserved** Project: All Format: |
| 12:0 | **Pipe_A_Horizontal_Blank_Start** Project: All<br>This 13-bit field specifies the Horizontal Blank Start position expressed in terms of the absolute pixel number relative to the horizontal active display start. The value programmed should be the HBLANK Start pixel position, where the first active pixel is considered position 0; the second active pixel is considered position 1, etc.<br>The number of clocks for both left and right borders need to be a multiple of two when driving the LVDS port in two channel mode. Horizontal blank should only start after the end of the horizontal active region.<br>The value loaded in the register would be equal to RightBorder+Active-1.<br>The border must be 0, so this register must always be programmed to the same value as the Horizontal Active. |

### 3.1.3 HSYNC_A—Pipe A Horizontal Sync Register

| HSYNC_A—Pipe A Horizontal Sync Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **60008h-6000Bh** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Size (in bits):** | 32 |

| Bit | Description |
|---|---|
| 31:29 | **Reserved** Project: All Format: MBZ |
| 28:16 | **Pipe_A_Horizontal_Sync_End**<br>Project: All<br>Default Value: 0b<br>This 13-bit field specifies the horizontal Sync End position expressed in terms of the absolute pixel number relative to the horizontal active display start. The value programmed should be the HSYNC End pixel position, where the first active pixel is considered position 0; the second active pixel is considered position 1, etc.<br>The number of clocks in the sync period needs to be a multiple of two when driving the LVDS port in two channel mode. This value should be greater than the horizontal sync start position and would be loaded with the Active+RightBorder+FrontPorch+Sync-1. |
| 15:13 | **Reserved** Project: All Format: MBZ |
| 12:0 | **Pipe_A_Horizontal_Sync_Start**<br>Project: All<br>Default Value: 0b<br>This 13-bit field specifies the horizontal Sync Start position expressed in terms of the absolute pixel number relative to the horizontal active display start. The value programmed should be the HSYNC Start pixel position, where the first active pixel is considered position 0; the second active pixel is considered position 1, etc. Note that when HSYNC Start is programmed equal to HBLANK Start, both HSYNC and HBLANK will be asserted on the same pixel clock. It should never be programmed to less than HBLANK start.<br>The number of cycles from the beginning of the line needs to be a multiple of two when driving the LVDS port in two channel mode. This register should not be less than the horizontal active end. This register should be loaded with the Active+RightBorder+FrontPorch-1. |

### 3.1.4 VTOTAL_A—Pipe A Vertical Total Register

| VTOTAL_A—Pipe A Vertical Total Register | | | |
|---|---|---|---|
| **Register Type:** | MMIO | | |
| **Address Offset:** | **6000Ch-6000Fh** | | |
| **Project:** | All | | |
| **Default Value:** | 00000000h | | |
| **Access:** | R/W | | |
| **Size (in bits):** | 32 | | |
| **Bit** | **Description** | | |
| 31:29 | **Reserved** | Project: All | Format: |
| 28:16 | **Pipe_A_Vertical_Total_Display_Lines** | Project: All | |
| | This 13 bit field provides Vertical Total up to 8192 lines encompassing the Vertical Active Display Lines, top/bottom border and retrace period. The value programmed should be the number of lines required minus one. Vertical total needs to be large enough to be greater than the sum of the vertical active, vertical border, and the vertical blank regions. The vertical counter is incremented on the leading edge of the horizontal sync. For interlaced display modes, this indicates the total number of lines in both fields. In interlaced modes, hardware automatically divides this number by 2 to get the number of lines in each field. | | |
| 15:12 | **Reserved** | Project: All | Format: |
| 11:0 | **Pipe_A_Vertical_Active_Display_Lines** | Project: All | |
| | This 12-bit field provides vertical active display resolutions up to 4096 lines. It should be programmed with the desired number of lines minus one. When using the internal panel fitting logic, the minimum vertical active area must be seven lines. For interlaced display modes, this indicates the total number of lines in both fields. In interlaced modes, hardware automatically divides this number by 2 to get the number of lines in each field. | | |

### 3.1.5 VBLANK_A—Pipe A Vertical Blank Register

| VBLANK_A—Pipe A Vertical Blank Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **60010h-60013h** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | R/W |
| **Size (in bits):** | 32 |

| Bit | Description |
|---|---|
| 31:29 | **Reserved** Project: All Format: |
| 28:16 | **Pipe_A_Vertical_Blank_End** Project: All<br>This 13-bit field specifies the Vertical Blank End position expressed in terms of the absolute Line number relative to the vertical active display start. The value programmed should be the VBLANK End line position, where the first active line is considered line 0, the second active line is considered line 1, etc. The end of vertical blank should be after the start of vertical blank and before or equal to the vertical total. This register should be loaded with the Vactive+BottomBorder+VBlank-1. For interlaced display modes, hardware automatically divides this number by 2 to get the vertical blank end in each field. It does not count the two half lines that get added when operating in modes with half lines.<br>The border must be 0, so this register must always be programmed to the same value as the Vertical Total. |
| 15:13 | **Reserved** Project: All Format: |
| 12:0 | **Pipe_A_Vertical_Blank_Start** Project: All<br>This 13-bit field specifies the Vertical Blank Start expressed in terms of the absolute line number relative to the vertical active display start. The value programmed should be the VBLANK Start line position, where the first active line is considered line 0, the second active line is considered line 1, etc. Minimum vertical blank size is required to be at least three lines. Blank should start after the end of active. This register is loaded with the Vactive+BottomBorder-1. For interlaced display modes, hardware automatically divides this number by 2 to get the vertical blank start in each field. It does not count the two half lines that get added when operating in modes with half lines.<br>The border must be 0, so this register must always be programmed to the same value as the Vertical Active. |

### 3.1.6 VSYNC_A—Pipe A Vertical Sync Register

| VSYNC_A—Pipe A Vertical Sync Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **60014h-60017h** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | R/W |
| **Size (in bits):** | 32 |

| Bit | Description |
|---|---|
| 31:29 | **Reserved** Project: All Format: |
| 28:16 | **Pipe_A_Vertical_Sync_End** Project: All<br>This 13-bit field specifies the Vertical Sync End position expressed in terms of the absolute Line number relative to the vertical active display start. The value programmed should be the VSYNC End line position, where the first active line is considered line 0, the second active line is considered line 1, etc. This register should be loaded with Vactive+BottomBorder+FrontPorch+Sync-1. For interlaced display modes, hardware automatically divides this number by 2 to get the vertical sync end in each field. It does not count the two half lines that get added when operating in modes with half lines. |
| 15:13 | **Reserved** Project: All Format: |
| 12:0 | **Pipe_A_Vertical_Sync_Start** Project: All<br>This 13-bit field specifies the Vertical Sync Start position expressed in terms of the absolute line number relative to the vertical active display start. The value programmed should be the VSYNC Start line position, where the first active line is considered line 0, the second active line is considered line 1, etc. This register would be loaded with Vactive+BottomBorder+FrontPorch-1. For interlaced display modes, hardware automatically divides this number by 2 to get the vertical sync start in each field. It does not count the two half lines that get added when operating in modes with half lines. |

### 3.1.7 PIPEASRC—Pipe A Source Image Size

| PIPEASRC—Pipe A Source Image Size | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **6001Ch-6001Fh** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | R/W |
| **Size (in bits):** | 32 |
| **Double Buffer Update Point:** | Start of vertical blank |

| Bit | Description |
|---|---|
| 31:28 | **Reserved** Project: All Format: MBZ |
| 27:16 | **Pipe_A_Horizontal_Source_Image_Size** Project: All<br>This 12-bit field specifies Horizontal source image size up to 4096. This determines the size of the image created by the display planes sent to the blender. The value programmed should be the source image size minus one.<br>It must represent a size that is a multiple of two (even numbers) when driving the LVDS port in two channel mode. This implies that for this mode, the value programmed will always be an odd number.<br>Except in the case of panel fitting internal or in an external device, this register field would be programmed to a value identical to the horizontal active. This is the only register of the timing registers that is allowed to be programmed while the pipe is enabled. |
| 15:12 | **Reserved** Project: All Format: MBZ |
| 11:0 | **Pipe_A_Vertical_Source_Image_Size** Project: All<br>This 12-bit field specifies the vertical source image size up to 4096 lines. This determines the size of the image created by the display planes sent to the blender. The value programmed should be the source image size minus one.<br>Except in the case of panel fitting internal or in an external device, this register field would be programmed to a value identical to the vertical active.<br>For interlaced display modes, hardware automatically divides this number by 2 to get the vertical source image size in each field. |

## 3.1.8 VSYNCSHIFT_A— Vertical Sync Shift Register

| VSYNCSHIFT_A— Vertical Sync Shift Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **60028h-6002Bh** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | R/W |
| **Size (in bits):** | 32 |
| **Trusted Type:** | 1 |

| Bit | Description |
|---|---|
| 31:13 | **Reserved** Project: All Format: |
| 12:0 | **Pipe_A_Second_Field_Vertical_Sync_Shift** Project: All<br>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.<br>This value will only be used if the PIPEACONF is programmed to an interlaced mode.<br>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:<br>(horizontal sync start - floor[horizontal total / 2]).<br>(use the actual horizontal sync start and horizontal total values and not the minus one values programmed into the registers)<br>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. |

## 3.1.9 Pipe A M/N Values

Calculation of TU is as follows:

For modes that divide into the link frequency evenly,

Active/TU = payload/capacity

Please note that this is the same ratio as data m/n:

Payload/capacity = dot clk * bytes per pixel / ls_clk * # of lanes

### 3.1.9.1 PipeADataM1— Pipe A Data M value 1

| PipeADataM1— Pipe A Data M value 1 | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **60030h-60033h** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | R/W |
| **Size (in bits):** | 32 |

This is the primary pipe data M value used for embedded DisplayPort and FDI. It is used in conjunction with the data N value 1. When switching between two refresh rates, both the M1/N1 data and link values and the M2/N2 link values must be programmed. Data M value 1 is used for the higher power M value setting.

| Bit | Description |
|---|---|
| 31 | **Reserved** Project: All Format: MBZ |
| 30:25 | **TU_Size** Project: All<br>This field is the size of the transfer unit for DP, minus one.<br>**Default value to program = 111111 (TU size of 64)** |
| 24 | **Reserved** Project: All Format: MBZ |
| 23:0 | **Pipe_A_Data_M_value** Project: All<br>This field is the m value for internal use of the DDA. Calculation of this value is as follows:<br>**Data m/n = dot clock * bytes per pixel / ls_clk * # of lanes**<br>Please note that in the DisplayPort specification, dot clock is referred to as strm_clk<br>In HDMI pixel-repeat modes, the pixel repeat factor needs to be taken into account. The calculation of this value would then be as follows:<br>Data M/N = (dot_clock * bytes_per_pixel) /<br>(ls_clk * num_lanes * pix_repeat_factor) |

### 3.1.9.2 PipeADataN1— Pipe A Data N value 1

| PipeADataN1— Pipe A Data N value 1 | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **60034h-60037h** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | R/W |
| **Size (in bits):** | 32 |

This is the primary pipe data N value used for embedded DisplayPort and FDI. It is used in conjunction with the data M value 1. When switching between two refresh rates, both the M1/N1 data and link values and the M2/N2 link values must be programmed. This value updates at the beginning of vblank. Data N value 1 is used for the higher power N value setting.

| Bit | Description |
|---|---|
| 31:24 | **Reserved** Project: All Format: MBZ |
| 23:0 | **Pipe_A_Data_N_value** Project: All<br>This field is the n value for internal use of the DDA. Calculation of this value is as follows:<br>**Data m/n = dot clock * bytes per pixel / ls_clk * # of lanes**<br>Please note that in the DisplayPort specification, dot clock is referred to as strm_clk<br><br>In HDMI pixel-repeat modes, the pixel repeat factor needs to be taken into account. The calculation of this value would then be as follows:<br><br>Data M/N = (dot_clock * bytes_per_pixel) /<br>(ls_clk * num_lanes * pix_repeat_factor) |

### 3.1.9.3 PipeADataM2— Pipe A Data M value 2

| PipeADataM2— Pipe A Data M value 2 | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **60038h-6003Bh** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | R/W |
| **Size (in bits):** | 32 |

This is the second pipe data M value used for embedded DisplayPort and FDI. It is used in conjunction with the data N value 2. When switching between two refresh rates, both the M1/N1 data and link values and the M2/N2 link values must be programmed. This value updates at the beginning of vblank. Data M value 2 is used for the lower power M value setting.

| Bit | Description |
|---|---|
| 31 | **Reserved** Project: All Format: MBZ |
| 30:25 | **TU_Size** Project: All<br>This field is the size of the transfer unit for DP, minus one.<br>**Default value to program = 111111 (TU size of 64)** |
| 24 | **Reserved** Project: All Format: MBZ |
| 23:0 | **Pipe_A_Data_M_value** Project: All<br>This field is the m value for internal use of the DDA. Calculation of this value is as follows:<br>**Data m/n = dot clock * bytes per pixel / ls_clk * # of lanes**<br>Please note that in the DisplayPort specification, dot clock is referred to as strm_clk<br>In HDMI pixel-repeat modes, the pixel repeat factor needs to be taken into account. The calculation of this value would then be as follows:<br>Data M/N = (dot_clock * bytes_per_pixel) /<br>(ls_clk * num_lanes * pix_repeat_factor) |

### 3.1.9.4 PipeADataN2— Pipe A Data N value 2

| PipeADataN2— Pipe A Data N value 2 | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **6003Ch-6003Fh** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | R/W |
| **Size (in bits):** | 32 |

This is the second pipe data N value used for DisplayPort and FDI. It is used in conjunction with the data N value 2. When switching between two refresh rates, both the M1/N1 data and link values and the M2/N2 link values must be programmed. This value updates at the beginning of vblank. Data N value 2 is used for the lower power N value setting.

| Bit | Description |
|---|---|
| 31:24 | **Reserved** Project: All Format: MBZ |
| 23:0 | **Pipe_A_Data_N_value** Project: All<br>This field is the n value for internal use of the DDA. Calculation of this value is as follows:<br>**Data m/n = dot clock * bytes per pixel / ls_clk * # of lanes**<br>Please note that in the DisplayPort specification, dot clock is referred to as strm_clk<br><br>In HDMI pixel-repeat modes, the pixel repeat factor needs to be taken into account. The calculation of this value would then be as follows:<br><br>Data M/N = (dot_clock * bytes_per_pixel) /<br>(ls_clk * num_lanes * pix_repeat_factor) |

### 3.1.9.5 PipeADPLinkM1— Pipe A Link M value 1

| PipeADPLinkM1— Pipe A Link M value 1 | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **60040h-60043h** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | R/W |
| **Size (in bits):** | 32 |

This is the primary link data M value used for embedded DisplayPort and FDI. It is used in conjunction with the link N value 1. When switching between two refresh rates, both the M1/N1 data and link values and the M2/N2 link values must be programmed. This value updates at the beginning of vblank. Link M value 1 is used for the higher power M value setting.

| Bit | Description |
|---|---|
| 31:24 | **Reserved** Project: All Format: MBZ |
| 23:0 | **Pipe_A_Link_M_value** Project: All<br>This field is the m value for external transmission in the Main Stream Attributes. Calculation of this value is as follows:<br>**Link m/n = pixel clk / ls_clk**<br>Please note that in the DisplayPort specification, pixel clk is referred to as strm_clk<br>In HDMI pixel-repeat modes, the pixel repeat factor needs to be taken into account. The calculation of this value would then be as follows:<br>Link M/N = dot_clock / (ls_clk * pix_repeat_factor) |

### 3.1.9.6 PipeADPLinkN1— Pipe A Link N value 1

| PipeADPLinkN1— Pipe A Link N value 1 | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **60044h-60047h** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | R/W |
| **Size (in bits):** | 32 |

This is the primary link data N value used for embedded DisplayPort and FDI. It is used in conjunction with the link N value 1. When switching between two refresh rates, both the M1/N1 data and link values and the M2/N2 link values must be programmed. This value updates at the beginning of vblank. Link N value 1 is used for the higher power N value setting.

| Bit | Description |
|---|---|
| 31:24 | **Reserved** Project: All Format: MBZ |
| 23:0 | **Pipe_A_Link_N_value** Project: All<br>This field is the n value for external transmission in the Main Stream Attributes and VB-ID. Calculation of this value is as follows (to be filled in):<br>**Link m/n = pixel clk / ls_clk**<br>Please note that in the DisplayPort specification, pixel clk is referred to as strm_clk<br>In HDMI pixel-repeat modes, the pixel repeat factor needs to be taken into account. The calculation of this value would then be as follows:<br>Link M/N = dot_clock / (ls_clk * pix_repeat_factor) |

### 3.1.9.7 PipeADPLinkM2— Pipe A Link M value 2

| PipeADPLinkM2— Pipe A Link M value 2 | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **60048h-6004Bh** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | R/W |
| **Size (in bits):** | 32 |

This is the secondary link data M value used for embedded DisplayPort and FDI. It is used in conjunction with the link N value 2. When switching between two refresh rates, both the M1/N1 data and link values and the M2/N2 link values must be programmed. This value updates at the beginning of vblank. Link M value 2 is used for the lower power M value setting.

| Bit | Description |
|---|---|
| 31:24 | **Reserved** Project: All Format: MBZ |
| 23:0 | **Pipe_A_Link_M_value** Project: All<br>This field is the m value for external transmission in the Main Stream Attributes. Calculation of this value is as follows:<br>**Link m/n = pixel clk / ls_clk**<br>Please note that in the DisplayPort specification, pixel clk is referred to as strm_clk<br>In HDMI pixel-repeat modes, the pixel repeat factor needs to be taken into account. The calculation of this value would then be as follows:<br>Link M/N = dot_clock / (ls_clk * pix_repeat_factor) |

### 3.1.9.8 PipeADPLinkN2— Pipe A Link N value 2

## PipeADPLinkN2— Pipe A Link N value 2

| | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **6004Ch-6004Fh** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | R/W |
| **Size (in bits):** | 32 |

This is the secondary link data N value used for embedded DisplayPort and FDI. It is used in conjunction with the link N value 1. When switching between two refresh rates, both the M1/N1 data and link values and the M2/N2 link values must be programmed. This value updates at the beginning of vblank. Link N value 2 is used for the lower power N value setting.

| Bit | Description |
|---|---|
| 31:24 | **Reserved** Project: All Format: MBZ |
| 23:0 | **Pipe_A_Link_N_value** Project: All<br>This field is the n value for external transmission in the Main Stream Attributes and VB-ID. Calculation of this value is as follows (to be filled in):<br>**Link m/n = pixel clk / ls_clk**<br>Please note that in the DisplayPort specification, pixel clk is referred to as strm_clk<br>In HDMI pixel-repeat modes, the pixel repeat factor needs to be taken into account. The calculation of this value would then be as follows:<br>Link M/N = dot_clock / (ls_clk * pix_repeat_factor) |

## 3.2 Pipe B Timing

### 3.2.1 HTOTAL_B—Pipe B Horizontal Total Register

| HTOTAL_B—Pipe B Horizontal Total Register | | | | | |
|---|---|---|---|---|---|
| **Register Type:** | MMIO | | | | |
| **Address Offset:** | **61000h-61003h** | | | | |
| **Project:** | All | | | | |
| **Default Value:** | 00000000h | | | | |
| **Access:** | R/W | | | | |
| **Size (in bits):** | 32 | | | | |
| **Bit** | **Description** | | | | |
| 31:29 | **Reserved** | Project: | All | Format: | MBZ |
| 28:16 | **Pipe_B_Horizontal_Total_Display_Clocks**<br>See pipe A description | | | Project: | All |
| 15:12 | **Reserved** | Project: | All | Format: | MBZ |
| 11:0 | **Pipe_B_Horizontal_Active_Display_Pixels**<br>See pipe A description | | | Project: | All |

### 3.2.2 HBLANK_B—Pipe B Horizontal Blank Register

| HBLANK_B—Pipe B Horizontal Blank Register | | | | | |
|---|---|---|---|---|---|
| **Register Type:** | MMIO | | | | |
| **Address Offset:** | **61004h-61007h** | | | | |
| **Project:** | All | | | | |
| **Default Value:** | 00000000h | | | | |
| **Access:** | R/W | | | | |
| **Size (in bits):** | 32 | | | | |
| **Bit** | **Description** | | | | |
| 31:29 | **Reserved** | Project: | All | Format: | |
| 28:16 | **Pipe_B_Horizontal_Blank_End**<br>See pipe A description | | | Project: | All |
| 15:13 | **Reserved** | Project: | All | Format: | |
| 12:0 | **Pipe_B_Horizontal_Blank_Start**<br>See pipe A description | | | Project: | All |

### 3.2.3 HSYNC_B—Pipe B Horizontal Sync Register

| HSYNC_B—Pipe B Horizontal Sync Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **61008h-6100Bh** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | R/W |
| **Size (in bits):** | 32 |

| Bit | Description |
|---|---|
| 31:29 | **Reserved** Project: All Format: |
| 28:16 | **Pipe_B_Horizontal_Sync_End** Project: All<br>See pipe A description |
| 15:13 | **Reserved** Project: All Format: |
| 12:0 | **Pipe_B_Horizontal_Sync_Start** Project: All<br>See pipe A description |

### 3.2.4 VTOTAL_B—Pipe B Vertical Total Register

| VTOTAL_B—Pipe B Vertical Total Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **6100Ch-6100Fh** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | R/W |
| **Size (in bits):** | 32 |

| Bit | Description |
|---|---|
| 31:29 | **Reserved** Project: All Format: |
| 28:16 | **Pipe_B_Vertical_Total_Display_Lines** Project: All<br>See pipe A description |
| 15:12 | **Reserved** Project: All Format: |
| 11:0 | **Pipe_B_Vertical_Active_Display_Lines** Project: All<br>See pipe A description |

### 3.2.5 VBLANK_B—Pipe B Vertical Blank Register

| VBLANK_B—Pipe B Vertical Blank Register | | | | |
|---|---|---|---|---|
| **Register Type:** | MMIO | | | |
| **Address Offset:** | **61010h-61013h** | | | |
| **Project:** | All | | | |
| **Default Value:** | 00000000h | | | |
| **Access:** | R/W | | | |
| **Size (in bits):** | 32 | | | |
| **Bit** | **Description** | | | |
| 31:29 | **Reserved** | Project: All | Format: | |
| 28:16 | **Pipe_B_Vertical_Blank_End**<br>See pipe A description | | Project: | All |
| 15:13 | **Reserved** | Project: All | Format: | |
| 12:0 | **Pipe_B_Vertical_Blank_Start**<br>See pipe A description | | Project: | All |

### 3.2.6 VSYNC_B—Pipe B Vertical Sync Register

| VSYNC_B—Pipe B Vertical Sync Register | | | | |
|---|---|---|---|---|
| **Register Type:** | MMIO | | | |
| **Address Offset:** | **61014h-61017h** | | | |
| **Project:** | All | | | |
| **Default Value:** | 00000000h | | | |
| **Access:** | R/W | | | |
| **Size (in bits):** | 32 | | | |
| **Bit** | **Description** | | | |
| 31:29 | **Reserved** | Project: All | Format: | |
| 28:16 | **Pipe_B_Vertical_Sync_End**<br>See pipe A description | | Project: | All |
| 15:13 | **Reserved** | Project: All | Format: | |
| 12:0 | **Pipe_B_Vertical_Sync_Start**<br>See pipe A description | | Project: | All |

### 3.2.7 PIPEBSRC—Pipe B Source Image Size

| PIPEBSRC—Pipe B Source Image Size | | | | | |
|---|---|---|---|---|---|
| **Register Type:** | MMIO | | | | |
| **Address Offset:** | **6101Ch-6101Fh** | | | | |
| **Project:** | All | | | | |
| **Default Value:** | 00000000h | | | | |
| **Access:** | R/W | | | | |
| **Size (in bits):** | 32 | | | | |
| **Double Buffer Update Point:** | Start of vertical blank | | | | |
| **Bit** | **Description** | | | | |
| 31:28 | **Reserved** | Project: | All | Format: | MBZ |
| 27:16 | **Pipe_B_Horizontal_Source_Image_Size**<br>See pipe A description | | | Project: | All |
| 15:12 | **Reserved** | Project: | All | Format: | MBZ |
| 11:0 | **Pipe_B_Vertical_Source_Image_Size**<br>See pipe A description | | | Project: | All |

### 3.2.8 VSYNCSHIFT_B— Vertical Sync Shift Register

| VSYNCSHIFT_B— Vertical Sync Shift Register | | | | | |
|---|---|---|---|---|---|
| **Register Type:** | MMIO | | | | |
| **Address Offset:** | **61028h-6102Bh** | | | | |
| **Project:** | All | | | | |
| **Default Value:** | 00000000h | | | | |
| **Access:** | R/W | | | | |
| **Size (in bits):** | 32 | | | | |
| **Bit** | **Description** | | | | |
| 31:13 | **Reserved** | Project: | All | Format: | |
| 12:0 | **Pipe_B_Second_Field_Vertical_Sync_Shift**<br>See pipe A description | | | Project: | All |

## 3.3 Pipe B M/N Values

### 3.3.1 PipeBDataM1— Pipe B Data M value 1

| PipeBDataM1— Pipe B Data M value 1 | | | | | |
|---|---|---|---|---|---|
| **Register Type:** MMIO<br>**Address Offset:** **61030h-61033h**<br>**Project:** All<br>**Default Value:** 00000000h<br>**Access:** R/W<br>**Size (in bits):** 32 | | | | | |
| See pipe A description | | | | | |
| **Bit** | **Description** | | | | |
| 31 | **Reserved** | Project: | All | Format: | MBZ |
| 30:25 | **TU_Size**<br>See pipe A description | | | Project: | All |
| 24 | **Reserved** | Project: | All | Format: | MBZ |
| 23:0 | **Pipe_B_Data_M_value**<br>See pipe A description | | | Project: | All |

### 3.3.2 PipeBDataN1— Pipe B Data N value 1

| PipeBDataN1— Pipe B Data N value 1 | | | | | |
|---|---|---|---|---|---|
| **Register Type:** MMIO<br>**Address Offset:** **61034h-61037h**<br>**Project:** All<br>**Default Value:** 00000000h<br>**Access:** R/W<br>**Size (in bits):** 32 | | | | | |
| See pipe A description | | | | | |
| **Bit** | **Description** | | | | |
| 31:24 | **Reserved** | Project: | All | Format: | MBZ |
| 23:0 | **Pipe_B_Data_N_value**<br>See pipe A description | | | Project: | All |

### 3.3.3 PipeBDataM2— Pipe B Data M value 2

| PipeBDataM2— Pipe B Data M value 2 | | | |
|---|---|---|---|
| **Register Type:** MMIO<br>**Address Offset:** **61038h-6103Bh**<br>**Project:** All<br>**Default Value:** 00000000h<br>**Access:** R/W<br>**Size (in bits):** 32 | | | |
| See pipe A description | | | |
| **Bit** | **Description** | | |
| 31 | **Reserved** Project: All | | Format: MBZ |
| 30:25 | **TU_Size**<br>See pipe A description | | Project: All |
| 24 | **Reserved** Project: All | | Format: MBZ |
| 23:0 | **Pipe_B_Data_M_value**<br>See pipe A description | | Project: All |

### 3.3.4 PipeBDataN2— Pipe B Data N value 2

| PipeBDataN2— Pipe B Data N value 2 | | | |
|---|---|---|---|
| **Register Type:** MMIO<br>**Address Offset:** **6103Ch-6103Fh**<br>**Project:** All<br>**Default Value:** 00000000h<br>**Access:** R/W<br>**Size (in bits):** 32 | | | |
| See pipe A description | | | |
| **Bit** | **Description** | | |
| 31:24 | **Reserved** Project: All | | Format: MBZ |
| 23:0 | **Pipe_B_Data_N_value**<br>See pipe A description | | Project: All |

### 3.3.5 PipeBDPLinkM1— Pipe B Link M value 1

| PipeBDPLinkM1— Pipe B Link M value 1 | | | | |
|---|---|---|---|---|
| **Register Type:** | MMIO | | | |
| **Address Offset:** | **61040h-61043h** | | | |
| **Project:** | All | | | |
| **Default Value:** | 00000000h | | | |
| **Access:** | R/W | | | |
| **Size (in bits):** | 32 | | | |
| See pipe A description | | | | |
| **Bit** | **Description** | | | |
| 31:24 | **Reserved** | Project: All | Format: MBZ | |
| 23:0 | **Pipe_B_Link_M_value**<br>See pipe A description | | Project: All | |

### 3.3.6 PipeBDPLinkN1— Pipe B Link N value 1

| PipeBDPLinkN1— Pipe B Link N value 1 | | | | |
|---|---|---|---|---|
| **Register Type:** | MMIO | | | |
| **Address Offset:** | **61044h-61047h** | | | |
| **Project:** | All | | | |
| **Default Value:** | 00000000h | | | |
| **Access:** | R/W | | | |
| **Size (in bits):** | 32 | | | |
| See pipe A description. | | | | |
| **Bit** | **Description** | | | |
| 31:24 | **Reserved** | Project: All | Format: MBZ | |
| 23:0 | **Pipe_B_Link_N_value**<br>See pipe A description | | Project: All | |

### 3.3.7 PipeBDPLinkM2— Pipe B Link M value 2

| PipeBDPLinkM2— Pipe B Link M value 2 | | | |
|---|---|---|---|
| **Register Type:** | MMIO | | |
| **Address Offset:** | **61048h-6104Bh** | | |
| **Project:** | All | | |
| **Default Value:** | 00000000h | | |
| **Access:** | R/W | | |
| **Size (in bits):** | 32 | | |
| See pipe A description | | | |
| **Bit** | **Description** | | |
| 31:24 | **Reserved** Project: All | | Format: MBZ |
| 23:0 | **Pipe_B_Link_M_value**<br>See pipe A description | | Project: All |

### 3.3.8 PipeBDPLinkN2— Pipe B Link N value 2

| PipeBDPLinkN2— Pipe B Link N value 2 | | | |
|---|---|---|---|
| **Register Type:** | MMIO | | |
| **Address Offset:** | **6104Ch-6104Fh** | | |
| **Project:** | All | | |
| **Default Value:** | 00000000h | | |
| **Access:** | R/W | | |
| **Size (in bits):** | 32 | | |
| See pipe A description | | | |
| **Bit** | **Description** | | |
| 31:24 | **Reserved** Project: All | | Format: MBZ |
| 23:0 | **Pipe_B_Link_N_value**<br>See pipe A description | | Project: All |

# 3.4 Embedded DP CTL & Aux Channel

## 3.4.1 DP_A—DisplayPort A Control Register

### DP_A—DisplayPort A Control Register

**Register Type:** MMIO
**Address Offset:** 64000h
**Project:** All
**Default Value:** 00000018h
**Access:** R/W
**Size (in bits):** 32

| Bit | Description |
|---|---|
| 31 | **DisplayPort_A_Enable**<br>Project: All<br>Default Value: 0b<br>Disabling this port will put it in its lowest power state. Port enable takes place on the Vblank after being written.<br>[DevSNB] If DisplayPort A and another port have been both enabled, then prior to disabling DisplayPort A the FDI PLL must be enabled in either FDI_TXA_CTL or FDI_TXB_CTL, then DisplayPort A can be disabled, and then the FDI PLL can be restored to its previous state. |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Disable | Disable and tristates the Display Port A interface | All |
| 1b | Enable | Enable. This bit enables the Display Port A interface. | All |

| Bit | Description |
|---|---|
| 30 | **Pipe_Select**<br>Project: All<br>Default Value: 0b<br>This bit determines from which display pipe the source data will originate. Pipe selection takes place on the Vblank after being written |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Pipe A | Pipe A | All |
| 1b | Pipe B | Pipe B | All |

## DP_A—DisplayPort A Control Register

29:28

**Link_training_pattern_enable**

Project: All

Default Value: 0b

These bits are used for link initialization as defined in the DisplayPort specification. 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 | |
| 11b | None | **Link not in training:** Send normal pixels | |

27:22

**Voltage_swing_Pre-emphasis_level_set_SNB**

Project: DevSNB

Default Value: 000000b

These bits are used for setting link voltage swing and pre-emphasis.

| Value | Name | Description | Project |
|---|---|---|---|
| 000000b | 400mv_0db/ 600mv_0db | 400mv 0dB / 600mv 0dB<br>Actual 0.5v transition and 0.5v non-transition | DevSNB |
| 000001b | 400mv_3.5db | 400mv 3.5dB<br>Actual 0.5v transition and 0.33v non-transition | DevSNB |
| 111010b | 400mv 6db / 600mv_6db | 400mv 6dB / 600mv 6dB<br>Actual 1v transition and 0.5v non-transition | DevSNB |
| 111001b | 600mv 3.5db / 800mv_3.5db | 600mv 3.5dB / 800mv 3.5dB<br>Actual 1v transition and 0.66v non-transition | DevSNB |
| 111000b | 800mv_0db / 1200mv_0db | 800mv 0dB / 1200mv 0dB<br>Actual 1v transition and 1v non-transition | DevSNB |
| Others | Reserved | Reserved | All |

## DP_A—DisplayPort A Control Register

21:19 **Port_Width_Selection**

Project: All

Default Value: 0b

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

| Value | Name | Description | Project |
|---|---|---|---|
| 000b | x1 | x1 Mode (Default) | All |
| 001b | x2 | x2 Mode. | All |
| 010b | Reserved | Reserved | All |
| 011b | x4 | x4 Mode. | All |
| 1XXb | Reserved | Reserved | All |

18 **Enhanced_Framing_Enable**

Project: All

Default Value: 0b

This bit selects enhanced framing.

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

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Disable | Enhanced framing disabled | All |
| 1b | Enable | Enhanced framing enabled. | All |

17:16 **DP_PLL_Frequency_Select**

Project: All

Default Value: 0b

| Value | Name | Description | Project |
|---|---|---|---|
| 00b | 270mhz | 270mhz | All |
| 01b | 162mhz | 162mhz. | All |
| 1Xb | Reserved | Reserved | All |

## DP_A—DisplayPort A Control Register

**15** **Port_reversal**

Project: All

Default Value: 0b

Enables lane reversal within the port: lane 0 mapped to lane 3, lane 1 mapped to lane 2, etc. Port reversal does not affect AUX channel lane mapping.

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

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Normal | Port not reversed | All |
| 1b | Reversed | Port reversed | All |

**14** **DP_PLL_enable**

Project: All

Default Value: 0b

This bit enables the DP PLL. Please note that software must wait for the PLL warmup cycle before enabling the port through bit 31 of this register.

**[DevILK]** Work around required when enabling DP PLL while a pipe is enabled going to FDI

1. Wait for the start of vertical blank on the enabled pipe going to FDI
2. Program DP PLL enable

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Disable | **DP PLL not enabled** | All |
| 1b | Enable | **DP PLL enabled** | All |

**13:8** **Reserved** Project: All Format: MBZ

**7** **Reserved**

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

**5** **Reserved**

**4:3** **Sync_Polarity**

Project: All

Default Value: 11b VS and HS are active high

Indicates the polarity of Hsync and Vsync to be transmitted in MSA.

| Value | Name | Description | Project |
|---|---|---|---|
| 00b | Low | VS and HS are active low (inverted) | All |
| 01b | VS_Low, HS_High | VS is active low (inverted), HS is active high | All |
| 10b | VS_High, HS_Low | VS is active high, HS is active low (inverted) | All |
| 11b | High | VS and HS are active high | |

| DP_A—DisplayPort A Control Register | |
|---|---|
| 2 | **Digital_Display_A_Detected**<br>Project: All<br>Default Value: 0b<br>Read-only bit indicating whether a digital display was detected during initialization. This bit is qualified with the Embedded DisplayPort A capability fuse.<br>**Value** / **Name** / **Description** / **Project**<br>0b / No Detect / Digital display not detected during initialization / All<br>1b / Detected / Digital display detected during initialization / All |
| 1:0 | **Reserved** Project: All Format: MBZ |

## 3.4.2 DPA_AUX_CH_CTL—Display Port A AUX Channel Control

| DPA_AUX_CH_CTL—Display Port A AUX Channel Control | |
|---|---|
| **Register Type:** MMIO<br>**Address Offset:** **64010h-64013h**<br>**Project:** All<br>**Default Value:** 00050000h<br>**Access:** R/W Special<br>**Size (in bits):** 32 | |
| **Bit** | **Description** |
| 31 | **Send/Busy** Project: All Format:<br>Setting this bit to a one initiates the transaction, when read this bit will be a 1 until the transmission completes. The transaction is completed when the response is received or when a timeout occurs. Do not write a 1 again until transaction completes. Writes of 0 will be ignored.<br>**Programming Notes**<br>Do not change any fields while Busy bit 31 is asserted. |
| 30 | **Done** Project: All<br>Access: R/W Clear<br>A sticky bit that indicates the transaction has completed. SW must write a 1 to this bit to clear the event |
| 29 | **Interrupt_on_Done** Project: All Format:<br>Enable an interrupt in the hotplug status register when the transaction completes or times out. |
| 28 | **Time_out_error** Project: All<br>Access: R/W Clear<br>A sticky bit that indicates the transaction has timed out. SW must write a 1 to this bit to clear the event. |

## DPA_AUX_CH_CTL—Display Port A AUX Channel Control

**27:26 Time_out_timer_value**

Project: All

Default Value: 0b

The time count depends on the 2X bit clock divider (bits 10:0) being programmed for 2MHz.

| Value | Name | Description | Project |
|---|---|---|---|
| 00b | 400us | 400us | All |
| 01b | 600us | 600us | All |
| 10b | 800us | 800us | All |
| 11b | 1600us | 1600us | All |

**25 Receive_error** Project: All

Access: R/W Clear

Default Value: 0b

A sticky bit that indicates that the data received was corrupted, not in multiples of a full byte, or more than 20 bytes. SW must write a 1 to this bit to clear the event.

**24:20 Message_Size** Project: All Format:

This field is used to indicate the total number bytes to transmit (including the header). It also indicates the number of bytes received in a transaction (including the header). This field is valid only when the done bit is set and timeout or receive error has not occurred. Sync/Stop are not part of the message or the message size.

Reads of this field will give the response message size.

The read value will not be valid while Busy bit 31 is asserted.

Message sizes of 0 or >20 are not allowed.

**19:16 Precharge_Time** Project: All

Default Value: 0101b 10 us precharge

Used to determine the precharge time for the Aux Channel drivers.

The value is the number of microseconds times 2 (assuming 2X bit clock divider programmed for 2MHz).

Example:

For 10us precharge, program 5 (10us/2us).

**15:11 Reserved**

**10:0 2X_Bit_Clock_divider** Project: All

Used to determine the 2X bit clock the Aux Channel logic runs on.

This value divides the input clock frequency down to 2X bit clock rate. The 2X bit clock rate is ideally 2MHz (0.5us). The input clock is cdclk.

Example:

For 300MHz input clock and desired 2MHz 2X bit clock, program 150 (300MHz/2MHz).

### 3.4.3 DPA_AUX_CH_DATA1—Display Port A AUX Data Register 1

| DPA_AUX_CH_DATA1—Display Port A AUX Data Register 1 | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **64014h-64017h** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | R/W Special |
| **Size (in bits):** | 32 |

The read value will not be valid while Busy bit 31 is asserted.

| Bit | Description |
|---|---|
| 31:0 | **AUX_CH_DATA1** Project: All Format:<br>The first DWord of the message. The MSbyte is transmitted first. Reads will give the response data after transaction complete. |

### 3.4.4 DPA_AUX_CH_DATA2—Display Port A AUX Data Register 2

| DPA_AUX_CH_DATA2—Display Port A AUX Data Register 2 | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **64018h-6401Bh** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | R/W Special |
| **Size (in bits):** | 32 |

The read value will not be valid while Busy bit 31 is asserted.

| Bit | Description |
|---|---|
| 31:0 | **AUX_CH_DATA2** Project: All Format:<br>The second DWord of the message. The MSbyte is transmitted first. Only used if the message size is greater than 4. Reads will give the response data after transaction complete |

### 3.4.5 DPA_AUX_CH_DATA3—Display Port A AUX Data Register 3

| DPA_AUX_CH_DATA3—Display Port A AUX Data Register 3 | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **6401Ch-6401Fh** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | R/W Special |
| **Size (in bits):** | 32 |
| The read value will not be valid while Busy bit 31 is asserted. | |

| Bit | Description |
|---|---|
| 31:0 | **AUX_CH_DATA3** Project: All Format:<br>The third DWord of the message. The MSbyte is transmitted first. Only used if the message size is greater than 8. Reads will give the response data after transaction complete. |

### 3.4.6 DPA_AUX_CH_DATA4—Display Port A AUX Data Register 4

| DPA_AUX_CH_DATA4—Display Port A AUX Data Register 4 | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **64020h-64023h** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | R/W Special |
| **Size (in bits):** | 32 |
| The read value will not be valid while Busy bit 31 is asserted. | |

| Bit | Description |
|---|---|
| 31:0 | **AUX_CH_DATA4** Project: All Format:<br>The fourth DWord of the message. The MSbyte is transmitted first. Only used if the message size is greater than 12. Reads will give the response data after transaction complete. |

### 3.4.7 DPA_AUX_CH_DATA5—Display Port A AUX Data Register 5

| DPA_AUX_CH_DATA5—Display Port A AUX Data Register 5 | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **64024h-64027h** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | R/W Special |
| **Size (in bits):** | 32 |

The read value will not be valid while Busy bit 31 is asserted.

| Bit | Description |
|---|---|
| 31:0 | **AUX_CH_DATA5** Project: All Format:<br>The fifth DWord of the message. The MSbyte is transmitted first. Only used if the message size is greater than 16. Reads will give the response data after transaction complete. |

## 3.5 Panel Fitter Control Registers

### 3.5.1 PF_WIN_POS—Panel Fitter Window Position

| PFA_WIN_POS—Panel Fitter A Window Position | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **68070h-68073h** |
| **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 PFA_WIN_SZ |

| Bit | Description |
|---|---|
| 31:29 | **Reserved** Project: All Format: MBZ |
| 28:16 | **XPOS** Project: All<br>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<br>The Y coordinate (in lines) of the upper left most pixel of the panel fitter display window.<br>LSB must be zero for interlaced modes. |

| PFB_WIN_POS—Panel Fitter B Window Position | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **68870h-68873h** |
| **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 PFB_WIN_SZ |

| Bit | Description |
|---|---|
| 31:29 | **Reserved** Project: All Format: MBZ |
| 28:16 | **XPOS** Project: All<br>See PFA description |
| 15:12 | **Reserved** Project: All Format: MBZ |
| 11:0 | **YPOS** Project: All<br>See PFA description |

## 3.5.2 PF_WIN_SZ—Panel Fitter Window Size

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 the window size arm PF registers for the pipe.

| PFA_WIN_SZ—Panel Fitter A Window Size | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **68074h-68077h** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | R/W |
| **Size (in bits):** | 32 |
| **Double Buffer Update Point:** | Start of vertical blank |

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

| PFB_WIN_SZ—Panel Fitter B Window Size | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | 68874h |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | R/W |
| **Size (in bits):** | 32 |
| **Double Buffer Update Point:** | Start of vertical blank |

| Bit | Description |
|---|---|
| 31:29 | **Reserved** Project: All Format: MBZ |
| 28:16 | **XSIZE** Project: All<br>See PFA descriptio |
| 15:12 | **Reserved** Project: All Format: MBZ |
| 11:0 | **YSIZE** Project: All<br>See PFA description |

## 3.5.3 PF_CTRL_1—Panel Fitter Control 1

| PFA_CTRL_1—Panel Fitter A Control 1 | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **68080h-68083h** |
| **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 PFA_WIN_SZ |

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.

| Bit | Description |
|---|---|
| 31 | **Enable_Pipe_Scaler**<br>Project: All<br>Default Value: 0b |
| 30 | **Reserved** |
| 29 | **Reserved** |
| 28 | **Reserved** |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Disable | Data bypasses the scaler | All |
| 1b | Enable | The scaler is enabled | All |

## PFA_CTRL_1—Panel Fitter A Control 1

**27** **VADAPT** Project: All

Puts the adaptive vertical filter into adaptive mode, intended for use in interlace output modes only.

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

**26:25** **VADAPT_MODE** Project: All

Puts the adaptive vertical filter into adaptive mode, intended for use in interlace output modes only.

| 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

Filter coefficient selection

| Value | Name | Description | Project |
|---|---|---|---|
| 00b | Programmed | Programmed Coefficients (Recommended) | All |
| 01b | Hardcoded Med | Hardcoded Coefficients for Medium 3x3 Filtering | All |
| 10b | Edge Enhance | Hardcoded Coefficients for Edge Enhancing 3x3 Filtering | All |
| 11b | Edge Soften | Hardcoded Coefficients for Edge Softening 3x3 Filtering | All |

**22** **CHR_PREF** Project: All

Chroma Pre-filter enable.

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Disable | Pre-filter disabled | All |
| 1b | Enable | Pre-filter enabled | All |

**21** **Reserved**

**20** **Reserved**

**19:0** **Reserved** Project: All Format: MBZ

## PFB_CTRL_1—Panel Fitter B Control 1

**Register Type:** MMIO
**Address Offset:** **68880h-68883h**
**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 PFB_WIN_SZ

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.

| Bit | Description |
|---|---|
| 31 | **Enable_Pipe_Scaler**<br>Project: All<br>Default Value: 0b<br><br>**Value** / **Name** / **Description** / **Project**<br>0b / Disable / Data bypasses the scaler / All<br>1b / Enable / The scaler is enabled / All |
| 30 | **Reserved** |
| 29 | **Reserved** |
| 28 | **Reserved** |
| 27 | **VADAPT** Project: All<br>Puts the adaptive vertical filter into adaptive mode, intended for use in interlace output modes only.<br><br>**Value** / **Name** / **Description** / **Project**<br>0b / Disable / Adaptive filtering disabled / All<br>1b / Enable / Adaptive filtering enabled / All |
| 26:25 | **VADAPT_MODE** Project: All<br>Puts the adaptive vertical filter into adaptive mode, intended for use in interlace output modes only.<br><br>**Value** / **Name** / **Description** / **Project**<br>00b / Least Adaptive / Least Adaptive (Recommended) / All<br>01b / Moderately Adaptive / Moderately Adaptive / All<br>10b / Reserved / Reserved / All<br>11b / Most Adaptive / Most Adaptive / All |

| PFB_CTRL_1—Panel Fitter B Control 1 | |
|---|---|
| 24:23 | **FILTER_SELECT** Project: All<br>Filter coefficient selection<br>**Value** / **Name** / **Description** / **Project**<br>00b / Programmed / Programmed Coefficients (Recommended) / All<br>01b / Hardcoded Med / Hardcoded Coefficients for Medium 3x3 Filtering / All<br>10b / Edge Enhance / Hardcoded Coefficients for Edge Enhancing 3x3 Filtering / All<br>11b / Edge Soften / Hardcoded Coefficients for Edge Softening 3x3 Filtering / All |
| 22 | **CHR_PREF** Project: All<br>Chroma Pre-filter enable.<br>**Value** / **Name** / **Description** / **Project**<br>0b / Disable / Pre-filter disabled / All<br>1b / Enable / Pre-filter enabled / All |
| 21 | **Reserved** |
| 20 | **Reserved** |
| 19:0 | **Reserved** Project: All Format: MBZ |

## 3.5.4 Panel Fitter Coefficient Registers

Coefficients for the panel fitter filters are stored in sign-exponent-mantissa format. The number of mantissa bit varies based on the filter. Two filter coefficients are stored in each dword, the tables below show the data packing in each of the words. Unused bits are considered reserved and should be written zero. The default value of all coefficient registers is 00000000h. Coefficients greater than 1.0 are only allowed in the center tap of the filter.

For RGB modes the Luma and Chroma filter coeffs are programmed with the same values.

| Panel Fitter Coefficient Definition | |
|---|---|
| **Project:** All | |
| **Bit** | **Description** |
| 15 | **Sign_bit**<br>Project: All<br>**Value** / **Name** / **Description** / **Project**<br>0b / Positive / Positive / All<br>1b / Negative / Negative / All |
| 14 | **Reserved** Project: All Format: MBZ |

<table>
<tr><th colspan="2">Panel Fitter Coefficient Definition</th></tr>
<tr><td>13:12</td><td>Exponent_bits<br>Project: All<br>The meaning of the exponent bits varies for center tap or non-center tap coefficients.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>00b</td><td>2 or 0.125</td><td>Center taps: 2 or mantissa is b.bbbbbb…<br>Non-center taps: 0.125 or mantissa is 0.000bbbbbbb</td><td>All</td></tr>
<tr><td>01b</td><td>1</td><td>1 or mantissa is 0.bbbbbbb…</td><td>All</td></tr>
<tr><td>10b</td><td>0.5</td><td>0.5 or mantissa is 0.0bbbbbbb…</td><td>All</td></tr>
<tr><td>11b</td><td>0.25</td><td>0.25 or mantissa is 0.00bbbbbbb…</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>Mantissa Project: All<br>Size of the mantissa varies based on the filter, but the MSB of the mantissa is always bit 11.<br>Center tap coefficients use all 9 bits of mantissa.<br>Non-center tap coefficients use only the upper 7 bits of mantissa and the lower 2 bits are ignored.</td></tr>
<tr><td>2:0</td><td>Reserved Project: All Format: MBZ</td></tr>
</table>

## 3.6 Panel Fitter Horizontal Coefficients

Coefficients are packed in the horizontal coefficient registers as follows (with the letter representing the tap and the number representing the coefficient set):

| Address | bits [31:16] | bits[15:0] |
|---|---|---|
| 68x00 | B0 | A0 |
| 68x04 | D0 | C0 |
| 68x08 | F0 | E0 |
| 68x0C | A1 | G0 |
| 68x10 | C1 | B1 |

etc….

### 3.6.1 PF_HFILTL_COEF—Panel Fitter Horizontal Luma/Red Coefficients

## PFA_HFILTL_COEF—Panel Fitter A Horizontal Luma/Red Coefficients

**Register Type:** MMIO
**Address Offset:** **68100h-681EFh**
**Project:** All
**Default Value:** 00000000h
**Access:** R/W
**Size (in bits):** 60x32

17 phases of 7 taps require 60 dwords

Center coefficient is 1.2.9

Other coefficients are 1.2.7

| DWord | Bit | Description | | | | |
|---|---|---|---|---|---|---|
| 0..59 | 31:16 | **Coefficient2** | Project: | All | Format: | Panel Fitter Coefficient Definition |
| | 15:0 | **Coefficient1** | Project: | All | Format: | Panel Fitter Coefficient Definition |

## PFB_HFILTL_COEF—Panel Fitter B Horizontal Luma/Red Coefficients

**Register Type:** MMIO
**Address Offset:** **68900h-689EFh**
**Project:** All
**Default Value:** 00000000h
**Access:** R/W
**Size (in bits):** 60x32

17 phases of 7 taps require 60 dwords

Center coefficient is 1.2.9

Other coefficients are 1.2.7

| DWord | Bit | Description | | | | |
|---|---|---|---|---|---|---|
| 0..59 | 31:16 | **Coefficient2** | Project: | All | Format: | Panel Fitter Coefficient Definition |
| | 15:0 | **Coefficient1** | Project: | All | Format: | Panel Fitter Coefficient Definition |

### 3.6.2 PF_HFILTC_COEF—Panel Fitter Horizontal Chroma/Green and Blue Coefficients

## PFA_HFILTC_COEF—Panel Fitter A Horizontal Chroma/Green and Blue Coefficients

**Register Type:** MMIO
**Address Offset:** **68200h-682EFh**
**Project:** All
**Default Value:** 00000000h
**Access:** R/W
**Size (in bits):** 60x32

17 phases of 7 taps require 60 dwords

Center coefficient is 1.2.9

Other coefficients are 1.2.7

| DWord | Bit | Description | | | | |
|---|---|---|---|---|---|---|
| 0..59 | 31:16 | **Coefficient2** | Project: | All | Format: | Panel Fitter Coefficient Definition |
| | 15:0 | **Coefficient1** | Project: | All | Format: | Panel Fitter Coefficient Definition |

## PFB_HFILTC_COEF—Panel Fitter B Horizontal Chroma/Green and Blue Coefficients

**Register Type:** MMIO
**Address Offset:** **68A00h-68AEFh**
**Project:** All
**Default Value:** 00000000h
**Access:** R/W
**Size (in bits):** 60x32

17 phases of 7 taps require 60 dwords

Center coefficient is 1.2.9

Other coefficients are 1.2.7

| DWord | Bit | Description | | | | |
|---|---|---|---|---|---|---|
| 0..59 | 31:16 | **Coefficient2** | Project: | All | Format: | Panel Fitter Coefficient Definition |
| | 15:0 | **Coefficient1** | Project: | All | Format: | Panel Fitter Coefficient Definition |

## 3.7 Panel Fitter Vertical Coefficients

Coefficients are packed in the vertical coefficient registers as follows (with the letter representing the tap and the number representing the coefficient set). When the vertical filter is in 3 line mode the three taps used are A, C & E, B & C must be programmed to zero in three line mode.

| Address | bits [31:16] | bits[15:0] |
|---|---|---|
| 68x00 | B0 | A0 |
| 68x04 | D0 | C0 |
| 68x08 | A1 | E0 |
| 68x0C | C1 | B1 |
| 68x10 | E1 | D1 |

etc….

### 3.7.1 PF_VFILTL_COEF—Panel Fitter Vertical Luma/Red Coefficients

#### PFA_VFILTL_COEF—Panel Fitter A Vertical Luma/Red Coefficients

| | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **68300h-683ABh** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | R/W |
| **Size (in bits):** | 43x32 |

17 phases of 5 taps require 43 dwords

Center coefficient is 1.2.9

Other coefficients are 1.2.7

| DWord | Bit | Description |
|---|---|---|
| 0..42 | 31:16 | **Coefficient2** Project: All Format: Panel Fitter Coefficient Definition |
| | 15:0 | **Coefficient1** Project: All Format: Panel Fitter Coefficient Definition |

## PFB_VFILTL_COEF—Panel Fitter B Vertical Luma/Red Coefficients

**Register Type:** MMIO
**Address Offset:** **68B00h-68BABh**
**Project:** All
**Default Value:** 00000000h
**Access:** R/W
**Size (in bits):** 43x32

17 phases of 5 taps require 43 dwords

Center coefficient is 1.2.9

Other coefficients are 1.2.7

| DWord | Bit | Description |
|---|---|---|
| 0..42 | 31:16 | **Coefficient2** Project: All Format: Panel Fitter Coefficient Definition |
| | 15:0 | **Coefficient1** Project: All Format: Panel Fitter Coefficient Definition |

### 3.7.2 PF_VFILTC_COEF—Panel Fitter Vertical Chroma/Green and Blue Coefficients

## PFA_VFILTC_COEF—Panel Fitter A Vertical Chroma/Green and Blue Coefficients

**Register Type:** MMIO
**Address Offset:** **68400h-684ABh**
**Project:** All
**Default Value:** 00000000h
**Access:** R/W
**Size (in bits):** 43x32

17 phases of 5 taps require 43 dwords

Center coefficient is 1.2.9

Other coefficients are 1.2.7

| DWord | Bit | Description |
|---|---|---|
| 0..42 | 31:16 | **Coefficient2** Project: All Format: Panel Fitter Coefficient Definition |
| | 15:0 | **Coefficient1** Project: All Format: Panel Fitter Coefficient Definition |

## PFB_VFILTC_COEF—Panel Fitter B Vertical Chroma/Green and Blue Coefficients

**Register Type:** MMIO
**Address Offset:** **68C00h-68CABh**
**Project:** All
**Default Value:** 00000000h
**Access:** R/W
**Size (in bits):** 43x32

17 phases of 5 taps require 43 dwords

Center coefficient is 1.2.9

Other coefficients are 1.2.7

| DWord | Bit | Description |
|---|---|---|
| 0..42 | 31:16 | **Coefficient2** Project: All Format: Panel Fitter Coefficient Definition |
| | 15:0 | **Coefficient1** Project: All Format: Panel Fitter Coefficient Definition |

# 4. Plane Controls (70000h–7FFFFh)

## 4.1 Display Pipeline A

### 4.1.1 PIPEA_DSL—Pipe A Display Scan Line

**PIPEA_DSL—Pipe A Display Scan Line**

**Register Type:** MMIO
**Address Offset:** **70000h-70003h**
**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 will 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**<br>Project: All<br>Default Value: 0<br>Provides read back of the current field being displayed on display pipe A.<br>**Value** / **Name** / **Description** / **Project**<br>0b / Odd / First field (odd field) / All<br>1b / Even / Second field (even field) / All |
| 30:13 | **Reserved** Project: All Format: |
| 12:0 | **Line_Counter_for_Display** Project: All<br>Provides read back of the display pipe A vertical line counter. This is an indication of the current display scan line to be used by software to synchronize with the display. |

## 4.1.2 PIPEA_SLC—Pipe A Display Scan Line Count Range Compare

### PIPEA_SLC—Pipe A Display Scan Line Count Range Compare

**Register Type:** MMIO
**Address Offset:** **70004h-70007h**
**Project:** All
**Default Value:** 00000000h
**Access:** R/W
**Size (in bits):** 32

**[DevSNB:D2+] Reserved.** The waid for scan line is programmed through DE_LOAD_SEL 0x4F100.

**[DevSNB:D2]** 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. Content locked display will adjust the total number of lines displayed.

| Bit | Description |
|---|---|
| 31 | **Inclusive/Exclusive**<br>Project: **DevILK**<br>Default Value: 0b<br>**Value** / **Name** / **Description** / **Project**<br>0b / Exclusive / Exclusive: outside of the range / All<br>1b / Inclusive / Inclusive: within the range / All |
| 30:29 | **Reserved** Project: **DevILK** Format: MBZ |
| 28:16 | **Start_Scan_Line_Number** Project: **DevILK** Format:<br>Range: 0..Vertical Total<br>This field specifies the starting scan line number of the Scan Line Window.<br>Scan line 0 is the first line of the display frame. |
| 15:13 | **Reserved** Project: **DevILK** Format: MBZ |
| 12:0 | **End_Scan_Line_Number** Project: **DevILK** Format:<br>Range: 0..Vertical Total<br>This field specifies the ending scan line number of the Scan Line Window.<br>Scan line 0 is the first line of the display frame. |
| 31:13 | **Reserved** Project: **DevSNB** Format: MBZ |
| 12:0 | **Scan_Line_Number**<br>Project: **Pre-DevSNB:D2**<br>Range 0..Vertical Total<br>This field specifies the scan line number on which to generate scan line interrupt and render response. |
| 12:0 | **Reserved** Project: **DevSNB:D2+** Format: MBZ |

### 4.1.3 PIPEACONF—Pipe A Configuration Register

## PIPEACONF—Pipe A Configuration Register

| | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **70008h-7000Bh** |
| **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 | **Pipe_A_Enable**<br>Project: All<br>Default Value: 0b<br>Setting this bit to the value of one, turns on pipe A. This must be done before any planes are enabled on this pipe. Changing it to a zero should only be done when all planes that are assigned to this pipe have been disabled. Turning the pipe enable bit off disables the timing generator in this pipe. Synchronization pulses to the display are not maintained if the timing generator is disabled. Pipe timing registers must contain valid values before this bit is enabled. |

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

| Bit | Description |
|---|---|
| 30 | **Pipe_State**<br>Project: All<br>Default Value: 0b<br>This 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. |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Disable | Disable State | All |
| 1b | Enable | Enable State | All |

| | |
|---|---|
| 29 | **Reserved** Project: All |
| 28:27 | **Reserved** |
| 26 | **Reserved** Project: DevSNB |

## PIPEACONF—Pipe A Configuration Register

25:24

**Pipe_A_Palette/Gamma_Unit_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 | Reserved | Reserved | All |

23:21

**Interlaced_Mode**

Project: All

Default Value: 0b

These bits are used for software control of interlaced behavior. They are updated immediately if the pipe is off, or in the vertical blank after programming if pipe is enabled.

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

| Value | Name | Description | Project |
|---|---|---|---|
| 000b | PF-PD | Progressive Fetch / Progressive display | All |
| 001b | PF-ID | Progressive Fetch / Interlaced display (HDMI) **Requires panel fitting to be enabled** | All |
| 010b | Reserved | Reserved | All |
| 011b | IF-ID | Interlaced Fetch / Interlaced display (programmable sync, normal interlaced) | All |
| 100b | IF-ID-DBL | Interlaced embedded panel with interlaced fetch (pixel doubling power savings mode, no PF enabled) | All |
| 101b | PF-ID-DBL | Interlaced embedded panel with progressive fetch (pixel doubling power savings mode with PF enabled) **Requires panel fitting to be enabled** | All |
| others | Reserved | Reserved | All |

## PIPEACONF—Pipe A Configuration Register

20

**Display_Power_Mode_Switch**

Project: All

Default Value: 0b

This bit is used for software to set the power saving progressive mode. The pipe enters or exits the power savings mode on the vblank after this bit is written. Please note that bits 17:16 of this register must be set to 00 in order for this bit to take effect.

If LVDS clockgating is disabled (bits 14, 30 set to '1' in 0xC2020), then clockgating must be temporarily enabled (bits 14, 30 cleared to '0') when toggling Display Power Mode Switch followed by wait of 2 vblanks and then disabled again.

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Progressive | Pipe is in progressive mode | All |
| 1b | Power save | Pipe is in power savings progressive mode | All |

19:18

**MSA_Timing_Delay**

Project: DevSNB

Default Value: 00b

This field selects the vertical blank line on which MSA is sent. It is intended for use with embedded DisplayPort panels that support sDRRS. The sDRRS timing switch shall occur on same line as the MSA.

| 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 |

19:18 **Reserved** Project: DevILK Format: MBZ

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 |

## PIPEACONF—Pipe A Configuration Register

**13** — **Color_Range_Select**

Project: All

Default Value: 0b

This bit is used to select the color range of outputs.

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Full | Apply full 0-$2^n$ - 1 color range to the output | All |
| 1b | CE | Apply CE color range to the output | All |

**12:11** — **Pipe_output_color_space_select**

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.

| Value | Name | Description | Project |
|---|---|---|---|
| 00b | RGB | RGB | All |
| 01b | YUV 601 | YUV 601 | All |
| 10b | YUV 709 | YUV 709 | All |
| 11b | Reserved | Reserved | All |

**10:9** — **Reserved** Project: DevSNB

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

**7:5** — **Bits_Per_Color**

Project: All

Default Value: 0b

This field selects the number of bits per color sent to a receiver device connected to this pipe. Color format takes place on the Vblank after being written. Color format change can be done independent of a pixel clock change.

Software should enable dithering in the pipe/port if selecting a pixel color depth higher or lower than the pixel color depth of the frame buffer.

| Value | Name | Description | Project |
|---|---|---|---|
| 000b | 8 bits | 8 bits per color | All |
| 001b | 10 bits | 10 bits per color | All |
| 010b | 6 bits | 6 bits per color | All |
| 011b | 12 bits | 12 bits per color | All |
| 1XXb | Reserved | Reserved | All |

## PIPEACONF—Pipe A Configuration Register

**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 only | All |
| 01b | ST1 | Spatio-Temporal 1 | All |
| 10b | ST2 | Spatio-Temporal 2 (testmode) | All |
| 11b | Temporal | Temporal only (testmode) | All |

**1** — **Reserved**

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

## 4.2 Display Pipeline A Counters and Timestamps

### 4.2.1 PIPEA_FRMCOUNT—Pipe A Frame Counter

| PIPEA_FRMCOUNT—Pipe A Frame Counter | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **70040h-70043h** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | Read Only |
| **Size (in bits):** | 32 |

| Bit | Description |
|---|---|
| 31:0 | **Pipe_Frame_Counter** Project: All Format:<br>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 frames. |

### 4.2.2 PIPEA_FLIPCOUNT—Pipe A Flip Counter

| PIPEA_FLIPCOUNT—Pipe A Flip Counter | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **70044h-70047h** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | Read Only |
| **Size (in bits):** | 32 |

| Bit | Description |
|---|---|
| 31:0 | **Pipe_Flip_Counter** Project: All Format:<br>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 any MMIO writes to the primary plane surface address. It rolls over back to 0 after 2^32 flips. |

### 4.2.3 PIPEA_FRMTIMESTAMP—Pipe A Frame Time Stamp

| PIPEA_FRMTIMESTAMP—Pipe A Frame Time Stamp | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **70048h-7004Bh** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | Read Only |
| **Size (in bits):** | 32 |

| Bit | Description |
|---|---|
| 31:0 | **Pipe_Frame_Time_Stamp** Project: All Format:<br>Provides read back of the display pipe frame time stamp. The time stamp value is sampled at every start of vertical blank. The TIMESTAMP register has information on the time stamp value. |

### 4.2.4 PIPEA_FLIPTIMESTAMP—Pipe A Flip Time Stamp

| PIPEA_FLIPTIMESTAMP—Pipe A Flip Time Stamp | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **7004Ch-7004Fh** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | Read Only |
| **Size (in bits):** | 32 |

| Bit | Description |
|---|---|
| 31:0 | **Pipe_Flip_Time_Stamp** Project: All Format:<br>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 any MMIO writes to the primary plane surface address. The TIMESTAMP register has information on the time stamp value. |

## 4.3 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.

### 4.3.1 TIMESTAMP_HI—Time Stamp High Value

| TIMESTAMP_HI—Time Stamp High Value | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **70070h-70073h** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | R/W Clear |
| **Size (in bits):** | 32 |

| Bit | Description |
|---|---|
| 31:0 | **TIMESTAMP_High** Project: All Format:<br>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. |

## 4.4 Display Pipeline B

### 4.4.1 PIPEB_DSL—Pipe B Display Scan Line

| PIPEB_DSL—Pipe B Display Scan Line | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **71000h-71003h** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | Read Only |
| **Size (in bits):** | 32 |

See Pipe A description

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31</td><td>Current_Field<br>Project: All<br>Default Value: 0b<br>Provides read back of the current field being displayed on display pipe B.
<table>
<tr><th>Value</th><th>Name</th><th>Description</th><th>Project</th></tr>
<tr><td>0b</td><td>First</td><td>First field (odd field)</td><td>All</td></tr>
<tr><td>1b</td><td>Second</td><td>Second field (even field)</td><td>All</td></tr>
</table></td></tr>
<tr><td>30:13</td><td>Reserved Project: All Format: MBZ</td></tr>
</table>

| PIPEB_DSL—Pipe B Display Scan Line | |
|---|---|
| 12:0 | **Line_Counter_for_Display** Project: All Format:<br>See pipe A description. |

## 4.4.2 PIPEB_SLC—Pipe B Display Scan Line Count Range Compare

| PIPEB_SLC—Pipe B Display Scan Line Count Range Compare | | | | | |
|---|---|---|---|---|---|
| **Register Type:** MMIO<br>**Address Offset:** **71004h-71007h**<br>**Project:** All<br>**Default Value:** 00000000h<br>**Access:** R/W<br>**Size (in bits):** 32 | | | | | |
| See Pipe A description | | | | | |
| **Bit** | **Description** | | | | |
| 31:13 | **Reserved** | Project: | **DevSNB** | Format: | MBZ |
| 12:0 | **Scan_Line_Number**<br>Project: **Pre-DevSNB:D2**<br>Range 0..Vertical Total<br>See pipe A description | | | | |
| 12:0 | **Reserved** | Project: | DevSNB:D2+ | Format: | MBZ |

### 4.4.3 PIPEBCONF—Pipe B Configuration Register

## PIPEBCONF—Pipe B Configuration Register

**Register Type:** MMIO
**Address Offset:** **71008h-7100Bh**
**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 | **Pipe_B_Enable**<br>Project: All<br>Default Value: 0b<br>Setting this bit to the value of one, turns on pipe B. This must be done before any planes are enabled on this pipe. Changing it to a zero should only be done when all planes that are assigned to this pipe have been disabled. Turning the pipe enable bit off disables the timing generator in this pipe. Synchronization pulses to the display are not maintained if the timing generator is disabled. Pipe timing registers must contain valid values before this bit is enabled. |
| 30 | **Pipe_State**<br>Project: All<br>Default Value: 0b<br>This 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. |
| 29 | **Reserved** Project: All Format: |
| 28:27 | **Reserved** |
| 26 | **Reserved** Project: DevSNB Format: |

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

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

## PIPEBCONF—Pipe B Configuration Register

25:24

**Pipe_B_Palette/Gamma_Unit_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 bit | 10-bit Precision Palette Mode | All |
| 10b | 12 bit | 12-bit Interpolated Gamma Mode | All |
| 11b | Reserved | Reserved | All |

23:21

**Interlaced_Mode**

Project: All

Default Value: 0b

These bits are used for software control of interlaced behavior. They are updated immediately if the pipe is off, or in the vertical blank after programming if pipe is enabled.

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

| Value | Name | Description | Project |
|---|---|---|---|
| 000b | PF-PD | Progressive Fetch / Progressive display | All |
| 001b | PF-ID | Progressive Fetch / Interlaced display (HDMI) **Requires panel fitting to be enabled** | All |
| 010b | Reserved | Reserved | All |
| 011b | IF-ID | Interlaced Fetch / Interlaced display (programmable sync, normal interlaced) | All |
| 100b | IF-ID-DBL | Interlaced embedded panel with interlaced fetch (pixel doubling power savings mode, no PF enabled) | All |
| 101b | PF-ID-DBL | Interlaced embedded panel with progressive fetch (pixel doubling power savings mode with PF enabled) **Requires panel fitting to be enabled in progressive mode** | All |
| 11Xb | Reserved | Reserved | All |

## PIPEBCONF—Pipe B Configuration Register

20

**Display_Power_Mode_Switch**

Project: All

Default Value: 0b

This bit is used for software to set the power saving progressive mode. The pipe enters or exits the power savings mode on the vblank after this bit is written. Please note that bits 17:16 of this register must be set to 00 in order for this bit to take effect.

If LVDS clockgating is disabled (bits 14, 30 set to '1' in 0xC2020), then clockgating must be temporarily enabled (bits 14, 30 cleared to '0') when toggling Display Power Mode Switch followed by wait of 2 vblanks and then disabled again.

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Progressive | Pipe is in progressive mode | All |
| 1b | Power Save | Pipe is in power savings progressive mode | All |

19:18

**MSA_Timing_Delay**

Project: DevSNB

Default Value: 00b

This field selects the vertical blank line on which MSA is sent. It is intended for use with embedded DisplayPort panels that support sDRRS. The sDRRS timing switch shall occur on same line as the MSA.

| 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 |

## PIPEBCONF—Pipe B Configuration Register

17:16 **Refresh_Rate_Power_Savings_Mode_Association**

Project: All

Default Value: 00b

These bits select how refresh rates are switched on pipe B. When they are set to anything other than 00, bits 23:21 of this register must be programmed to 000. Switching between 01 and 10 settings directly is not allowed. Software is responsible for enabling this mode only for integrated display panels that support corresponding mode. The refresh rate hardware control register has additional settings for hardware controlled refresh rate switching.

| Value | Name | Description | Project |
|---|---|---|---|
| 00b | None | No dynamic refresh rate change enabled. Software control through bits 23:21 only | All |
| 01b | PTP | Progressive-to-progressive refresh rate change enabled. For the CPU and PCH, link and data M and N 1 values are used for high power settings, M and N 2 values for low power settings. For the PCH, pixel clock FPB0 values are used for high power settings, FPB1 values for low power settings. | All |
| 10b | PTI | Progressive-to-interlaced refresh rate change enabled. Pixel clock value does not change in this case. However, data and link M values in the CPU and PCH (as appropriate) are divided by 2 (shifted right by 1 bit). If scaling is enabled, the planes will fetch progressive data which will be interlaced by the panel fitter. If scaling is disabled, the planes will fetch interlaced data, reducing the amount of data fetched | All |
| 11b | Reserved | Reserved | All |

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 |

## PIPEBCONF—Pipe B Configuration Register

13

**Color_Range_Select**

Project: All

Default Value: 0b

This bit is used to select the color range of outputs.

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Full | Apply full 0-$2^n$ - 1 color range to the output | All |
| 1b | CE | Apply CE color range to the output | All |

12:11

**Pipe_output_color_space_select**

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.

| Value | Name | Description | Project |
|---|---|---|---|
| 00b | RGB | RGB | All |
| 01b | YUV 601 | YUV 601 | All |
| 10b | YUV 709 | YUV 709 | All |
| 11b | Reserved | Reserved | All |

10:9

**Reserved** Project: DevSNB

8

**Reserved** Project: All Format: MBZ

7:5

**Bits_Per_Color**

Project: All

Default Value: 0b

This field selects the number of bits per color sent to a receiver device connected to this pipe. Color format takes place on the Vblank after being written. Color format change can be done independent of a pixel clock change.

Software should enable dithering in the pipe/port if selecting a pixel color depth higher or lower than the pixel color depth of the frame buffer.

For further details on Display Port fixed frequency programming to accommodate these formats refer to “DP Frequency Programming” in DPLL section of Bspec.

| Value | Name | Description | Project |
|---|---|---|---|
| 000b | 8 bits | 8 bits per color | All |
| 001b | 10 bits | 10 bits per color | All |
| 010b | 6 bits | 6 bits per color | All |
| 011b | 12 bits | 12 bits per color | All |
| 1XXb | Reserved | Reserved | All |

**PIPEBCONF—Pipe B Configuration Register**

| | |
|---|---|
| 4 | **Dithering_enable**<br>Project: All<br>Default Value: 0b<br>This bit enables dithering |

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

| | |
|---|---|
| 3:2 | **Dithering_type**<br>Project: All<br>Default Value: 0b<br>Security: Test<br>These bits select dithering type. |

| Value | Name | Description | Project |
|---|---|---|---|
| 00b | Spatial | Spatial only | All |
| 01b | ST1 | Spatio-Temporal 1 | All |

| | |
|---|---|
| 1 | **Reserved** |
| 0 | **Reserved** Project: All Format: MBZ |

# 4.5 Display Pipeline B Counters and Timestamps

## 4.5.1 PIPEB_FRMCOUNT—Pipe B Frame Counter

**PIPEB_FRMCOUNT—Pipe B Frame Counter**

**Register Type:** MMIO
**Address Offset:** **71040h-71043h**
**Project:** All
**Default Value:** 00000000h
**Access:** Read Only
**Size (in bits):** 32

| Bit | Description |
|---|---|
| 31:0 | **Pipe_Frame_Counter** Project: All Format:<br>See Pipe A description |

### 4.5.2 PIPEB_FLIPCOUNT—Pipe B Flip Counter

| PIPEB_FLIPCOUNT—Pipe B Flip Counter | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **71044h-71047h** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | Read Only |
| **Size (in bits):** | 32 |

| Bit | Description |
|---|---|
| 31:0 | **Pipe_Flip_Counter** Project: All Format:<br>See Pipe A description |

### 4.5.3 PIPEB_FRMTIMESTAMP—Pipe B Frame Time Stamp

| PIPEB_FRMTIMESTAMP—Pipe B Frame Time Stamp | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **71048h-7104Bh** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | Read Only |
| **Size (in bits):** | 32 |

| Bit | Description |
|---|---|
| 31:0 | **Pipe_Frame_Time_Stamp** Project: All Format:<br>See Pipe A description |

### 4.5.4 PIPEB_FLIPTIMESTAMP—Pipe B Flip Time Stamp

| PIPEB_FLIPTIMESTAMP—Pipe B Flip Time Stamp | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **7104Ch-7104Fh** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | Read Only |
| **Size (in bits):** | 32 |

| Bit | Description |
|---|---|
| 31:0 | **Pipe_Flip_Time_Stamp** Project: All Format:<br>See Pipe A description |

## 4.6 Cursor A Plane Control Registers

The CURACNTR active register will be updated on the vertical blank or when pipe is disabled, after the CURABASE or CURAPOPUPBASE trigger register is written, or when cursor A is not yet enabled – thus providing an atomic update of the cursor A control and base address registers.

### 4.6.1 CURACNTR—Cursor A Control Register

**CURACNTR—Cursor A Control Register**

| | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **70080h-70083h** |
| **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 CURABASE or CURAVGAPOPUPBASE |

For Hi-res modes Cursor A is connected to pipe A only. For VGA popup it follows the VGA pipe select.

| Bit | Description |
|---|---|
| 31:28 | **Reserved** Project: All Format: |
| 27 | **Popup_Cursor_Enabled**<br>Project: All<br>Default Value: 0b<br>This bit should be turned on when using Cursor A as a popup cursor. When in popup mode, hardware interprets the cursor base address as a physical address instead of a graphics address. |
| 26 | **Cursor_Gamma_Enabled**<br>Project: All<br>Default Value: 0b<br>This bit only has an effect when using the cursor in a non-VGA mode. In VGA pop-up operation, the cursor data will always bypass the gamma (palette) unit. |
| 25 | **Reserved** Project: All Format: |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Hi-Res | Cursor A is hi-res | All |
| 1b | VGA | Cursor A is popup | All |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Bypass | Cursor pixel data bypasses gamma correction or palette | All |
| 1b | Gamma | Cursor pixel data is gamma to be corrected in the pipe | All |

## CURACNTR—Cursor A Control Register

**24** — **Pipe_Color_Space_Conversion_Enable**

Project: All

Default Value: 0b

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

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

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

**15** — **180°_Rotation**

Project: All

Default Value: 0b

This mode causes the cursor to be rotated 180°. Only 32 bits per pixel cursors can be rotated. This field must be zero when the cursor format is 2 bits per pixel.

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | None | No rotation | All |
| 1b | 180 | 180° Rotation of 32 bit per pixel cursors | All |

**14** — **Trickle_Feed_Enable**

Project: DevSNB

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 |

**13:6** — **Reserved** Project: All Format:

**5** — **Cursor_Mode_Select[5]**

Project: All

Default Value: 0b

**Cursor Mode Select**

| Bit 5 | Bits 2:0 | Mode |
|---|---|---|
| 0 | 000 | Cursor is disabled. This is the default after reset. When the cursor register value changes from enabled to disabled, the cursor will stop fetching data at the following VBLANK event.<br>The cursor enable can be overridden by the pipe cursor disable bit. The value of these bits do not change when disabled by the pipe cursor disable bit. |
| 0 | 001 | Reserved |

| | | |
|---|---|---|
| 0 | 010 | 128 x 128 32bpp AND/INVERT (not available for VGA use)<br>See description off 64 x 64 32bpp AND/INVERT format for byte usage |
| 0 | 011 | 256 x 256 32bpp AND/INVERT (not available for VGA use)<br>See description off 64 x 64 32bpp AND/INVERT format for byte usage |
| 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 (not available for VGA use)<br>For each pixel:<br>Least significant three bytes provides cursor RGB 888 color information<br>Most Significant Byte:<br>All Ones: Opaque, show the cursor color<br>All Zeros: Transparent (color must also equal zero)<br>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) (not available for VGA use) |
| 1 | 011 | 256 x 256 32bpp ARGB (8:8:8:8 MSB-A:R:G:B) (not available for VGA use) |
| 1 | 100 | 64 x 64 32bpp AND/XOR (not available for VGA use)<br>For each pixel:<br>Least significant three bytes provides cursor RGB 888 color information<br>Most Significant Byte:<br>All Ones: Opaque, show the cursor color<br>All Zeros: Transparent (color must also equal zero)<br>Other: XOR the cursor color with the underlying display pixel data |
| 1 | 101 | 128 x 128 32bpp AND/XOR (not available for VGA use)<br>See description off 64 x 64 32bpp AND/XOR format for byte usage |
| 1 | 110 | 256 x 256 32bpp AND/XOR (not available for VGA use)<br>See description off 64 x 64 32bpp AND/XOR format for byte usage |
| 1 | 111 | 64 x 64 32bpp ARGB (8:8:8:8 MSB-A:R:G:B) (not available for VGA use) |

| | |
|---|---|
| 4:3 | **Reserved** Project: All Format: |
| 2:0 | **Cursor_Mode_Select[2:0]**<br>Project: All<br>Default Value: 0b<br>These three bits together with bit 5 select the mode for cursor as shown in the table described in bit 5. |

## 4.6.2 CURABASE—Cursor A Base Address Register

| CURABASE—Cursor A Base Address Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **70084h-70087h** |
| **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 CURA registers**

This register is only used when cursor A is in the hi-res mode. In VGA popup mode CURAVGAPOPUPBASE 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_Address[31:12]**<br>Project: All<br>Address: GraphicsAddress[31:12]<br>This field specifies bits 31:12 of the graphics address of the base of the cursor for hi-res mode. The physical address used for VGA popup cursor is in the CURAVGAPOPUPBASE register.<br>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 change, hardware will internally offset to start from the last pixel of the last line of the cursor. |
| 11:0 | **Reserved** Project: All Format: |

### 4.6.3 CURAPOS—Cursor A Position Register

| CURAPOS—Cursor A Position Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **70088h-7008Bh** |
| **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. The origin of the cursor position is always the upper left corner of the active image for the display pipe that the cursor is assigned.

| Bit | Description |
|---|---|
| 31 | **Cursor_Y-Position_Sign_Bit** Project: All<br>This bit provides the sign bit of a signed 13-bit value that specifies the horizontal position of cursor. For normal high resolution display modes, the cursor must have at least a single pixel positioned over the active screen. For use as a VGA Popup, the entire cursor must be positioned over the active area of the VGA image. |
| 30:28 | **Reserved** Project: All Format: MBZ |
| 27:16 | **Cursor_Y-Position_Magnitude_Bits** Project: All<br>This register provides the magnitude bits of a signed 12-bit value that specifies the vertical position of cursor. The sign bit of this value is provided by bit 31of this register. For use as a VGA Popup, the entire cursor must be positioned over the active area of the VGA image. Enabling the border in VGA (VGA Border Enable bit in the VGA Config register) includes the border in what is considered the “active area”.<br>When performing 180° rotation, this field specifies the vertical position of the lower right corner relative to the end of the active video area in the unrotated orientation |
| 15 | **Cursor_X-Position_Sign_Bit** Project: All<br>This bit provides the sign bit of a signed 13-bit value that specifies the horizontal position of cursor. For normal high resolution display modes, the cursor must have at least a single pixel positioned over the active screen. For use as a VGA Popup, the entire cursor must be positioned over the active area of the VGA image. Enabling the border in VGA (VGA Border Enable bit in the VGA Config register) includes the border in what is considered the “active area”. |
| 14:12 | **Reserved** Project: All Format: MBZ |
| 11:0 | **Cursor_X-Position_Magnitude_Bits** Project: All<br>These 12 bits provide the signed 13-bit value that specifies the horizontal position of cursor. The sign bit is provided by bit 15 of this register.<br>When performing 180° rotation, this field specifies the horizontal position of the lower right corner relative to the end of the active video area in the unrotated orientation. |

### 4.6.4 CURAVGAPOPUPBASE—Cursor A VGA Popup Base Address Register

| CURAVGAPOPUPBASE—Cursor A VGA Popup Base Address Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **7008Ch-7008Fh** |
| **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 CURA registers**

This register is only used when cursor A is in the VGA popup mode. In hi-res mode CURABASE 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_VGA_Popup_Base_Address[31:12]**<br>Project: All<br>Address: PhysicalAddress[31:12]<br>This field specifies bits 31:12 of the physical address of the base of the cursor for VGA popup mode. The graphics address used for hi-res cursor is in the CURABASE register.<br>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_VGA_Popup_Base_Address_MSBs[38:32]**<br>Project: All<br>Address: PhysicalAddress[38:32]<br>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. |

## 4.6.5 CURAPALET—Cursor A Palette registers

| Cursor Palette Format | | |
|---|---|---|
| **Project:** | All | |
| **Bit** | **Description** | |
| 31:24 | **Reserved** Project: All Format: MBZ | |
| 23:16 | **Red_or_Y_Value** Project: All Format:<br>These registers specify the cursor palette. RGB data is full range unsigned numbers. YUV data will be unsigned for the Y and excess 128 notation for the UV values. The data can be pre-gamma corrected and bypass the gamma correction logic or passed through the gamma corrector. | |
| 15:8 | **Green_or_U_Value** Project: All Format: | |
| 7:0 | **Blue_or_V_Value** Project: All Format: | |

| CURAPALET—Cursor A Palette registers | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **70090h-7009Fh** |
| **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 one of 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 cursor palette provides color information when using one of 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 | 2 color | 3color | 4color |
|---|---|---|---|
| 00 | palette 0 | palette 0 | palette 0 |
| 01 | palette 1 | palette 1 | palette 1 |
| 10 | transparent | transparent | palette 2 |
| 11 | invert destination<br>(palette 3 all 1s) | palette 3 | palette 3 |

Palette 3 must be programmed with all 1s for invert destination.

| DWord | Bit | Description |
|---|---|---|
| 0 | 31:0 | **CURAPALET0** Project: All Format: Cursor Palette Format |
| 1 | 31:0 | **CURAPALET1** Project: All Format: Cursor Palette Format |

| CURAPALET—Cursor A Palette registers | | | | | | |
|---|---|---|---|---|---|---|
| 2 | 31:0 | **CURAPALET2** | Project: | All | Format: | Cursor Palette Format |
| 3 | 31:0 | **CURAPALET3** | Project: | All | Format: | Cursor Palette Format |

### 4.6.6 CURASURFLIVE—Cursor A Live Surface Base Address

| CURASURFLIVE—Cursor A Live Surface Base Address | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **700ACh-700AFh** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | Read Only |
| **Size (in bits):** | 32 |

| Bit | Description |
|---|---|
| 31:0 | **Cursor_A_Live_Surface_Base_Address** Project: All Format:<br>This gives the live value of the surface base address as being currently used for the plane. |

## 4.7 Cursor B Plane Control Registers

The CURBCNTR active register will be updated on the vertical blank or when pipe is disabled, after the CURBBASE trigger register is written, or when cursor B is not yet enabled – thus providing an atomic update of the cursor B control and base address registers.

### 4.7.1 CURBCNTR—Cursor B Control Register

## CURBCNTR—Cursor B Control Register

**Register Type:** MMIO
**Address Offset:** **700C0h-700C3h**
**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 CURBBASE

Cursor B is connected to pipe B only.

| Bit | Description |
|---|---|
| 31:27 | **Reserved** Project: All Format: MBZ |
| 26 | **Cursor_Gamma_Enable**<br>Project: All<br>Default Value: 0b<br><br>**Value** / **Name** / **Description** / **Project**<br>0b / Bypass / Cursor pixel data bypasses gamma correction / All<br>1b / Corrected / Cursor pixel data is gamma to be corrected / All |
| 25 | **Reserved** Project: All Format: |
| 24 | **Pipe_Color_Space_Conversion_Enable**<br>Project: All<br>Default Value: 0b<br>This bit enables pipe color space conversion for the cursor pixel data. CSC mode in the pipe CSC registers must be set to match the format of the cursor pixel data<br><br>**Value** / **Name** / **Description** / **Project**<br>0b / Bypass / Cursor pixel data bypasses the pipe color space conversion logic / All<br>1b / pass / Cursor pixel data passes through the pipe color space conversion logic. / All |
| 23:16 | **Reserved** Project: All Format: |
| 15 | **180°_Rotation**<br>Project: All<br>Default Value: 0b<br>This mode causes the cursor to be rotated 180°. Only 32 bits per pixel cursors can be rotated. This field must be zero when the cursor format is 2 bits per pixel.<br><br>**Value** / **Name** / **Description** / **Project**<br>0b / None / No rotation / All<br>1b / 180 / 180° Rotation of 32 bit per pixel cursors / All |

## CURBCNTR—Cursor B Control Register

| | |
|---|---|
| 14 | **Trickle_Feed_Enable**<br>Project: DevSNB<br>Default Value: 0b<br><br>**Value** / **Name** / **Description** / **Project**<br>0b / Enable / Trickle Feed Enabled - Data requests are sent whenever there is space in the Display Data Buffer. / All<br>1b / Disable / Trickle Feed Disabled - Data requests are sent in bursts. / All |
| 13:6 | **Reserved** Project: All Format: MBZ |
| 5 | **Cursor_Mode_Select** Project: All Format:<br>**Cursor Mode Select**<br>Defined in CURACNTR—Cursor A Control Register Bit 5. |
| 4:3 | **Reserved** Project: All Format: |
| 2:0 | **Cursor_Mode_Select** Project: All Format:<br>These three bits together with bit 5 select the mode for cursor as shown in CURACNTR—Cursor A Control Register Bit 5. |

## 4.7.2 CURBBASE—Cursor B Base Address Register

## CURBBASE—Cursor B Base Address Register

| | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **700C4h-700C7h** |
| **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 CURB registers

This register specifies the graphics memory address at which the cursor image data is located.

| Bit | Description |
|---|---|
| 31:12 | **Cursor_Base_Address[31:12]**<br>Project: All<br>Address: GraphicsAddress[31:12]<br>This register specifies the graphics address of the cursor. It also acts as a trigger event to force the update of active registers on the next display event.<br>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 change, hardware will internally offset to start from the last pixel of the last line of the cursor. |
| 11:0 | **Reserved** Project: All Format: MBZ |

### 4.7.3 CURBPOS—Cursor B Position Register

| CURBPOS—Cursor B Position Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **700C8h-700CBh** |
| **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. The origin of the cursor position is always the upper left corner of the active image for the display pipe that the cursor is assigned.

| Bit | Description |
|---|---|
| 31 | **Cursor_Y-Position_Sign_Bit** Project: All<br>This bit provides the sign bit of a signed 13-bit value that specifies the horizontal position of cursor. For normal high resolution display modes, the cursor must have at least a single pixel positioned over the active screen. |
| 30:28 | **Reserved** Project: All Format: MBZ |
| 27:16 | **Cursor_Y-Position_Magnitude_Bits** Project: All<br>This register provides the magnitude bits of a signed 13-bit value that specifies the vertical position of cursor. The sign bit of this value is provided by bit 31of this register.<br>When performing 180° rotation, this field specifies the vertical position of the lower right corner relative to the end of the active video area in the unrotated orientation. |
| 15 | **Cursor_X-Position_Sign_Bit** Project: All<br>This bit provides the sign bit of a signed 13-bit value that specifies the horizontal position of cursor. For normal high resolution display modes, the cursor must have at least a single pixel positioned over the active screen. |
| 14:12 | **Reserved** Project: All Format: MBZ |
| 11:0 | **Cursor_X-Position_Magnitude_Bits** Project: All<br>These 12 bits provide the signed 13-bit value that specifies the horizontal position of cursor. The sign bit is provided by bit 15 of this register.<br>When performing 180° rotation, this field specifies the vertical position of the lower right corner relative to the end of the active video area in the unrotated orientation. |

### 4.7.4 CURBPALET—Cursor B Palette registers

| CURBPALET—Cursor B Palette registers | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **700D0h-700DFh** |
| **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 one of 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 cursor palette provides color information when using one of 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. See Cursor A palette usage table.

| DWord | Bit | Description |
|---|---|---|
| 0 | 31:0 | **CURBPALET0** Project: All Format: Cursor Palette Format |
| 1 | 31:0 | **CURBPALET1** Project: All Format: Cursor Palette Format |
| 2 | 31:0 | **CURBPALET2** Project: All Format: Cursor Palette Format |
| 3 | 31:0 | **CURBPALET3** Project: All Format: Cursor Palette Format |

### 4.7.5 CURBSURFLIVE—Cursor B Live Surface Base Address Register

| CURBSURFLIVE—Cursor B Live Surface Base Address Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **700ECh-700Efh** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | Read Only |
| **Size (in bits):** | 32 |

| Bit | Description |
|---|---|
| 31:0 | **Cursor_Live_Surface_Base_Address** Project: All Format:<br>This gives the live value of the surface base address as being currently used for the plane. |

## 4.8 Primary A Plane Control

The DSPACNTR and DSPASTRIDE active registers will be updated on the vertical blank or when pipe is disabled, after the DSPASURF trigger register is written, or when the primary A is not yet enabled – thus providing an atomic update of the primary A control, stride, and base address registers.

### 4.8.1.1 DSPACNTR—Primary A Control Register

**DSPACNTR—Primary A Control Register**

| | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **70180h-70183h** |
| **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 DSPASURF |

Primary A Plane is connected to pipe A only.

| Bit | Description |
|---|---|
| 31 | **Primary_Plane_Enable** Project: All Format: Enable<br>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. The display pipe must be enabled to enable the plane. When in Self Refresh Big FIFO mode, write to this register to enable the plane will be internally buffered and delayed while Big FIFO mode is exiting. |
| 30 | **Gamma_Enable**<br>Project: All<br>Default Value: 0b<br>This bit should only be changed after the plane has been disabled. It controls the bypassing of the display pipe gamma unit for the plane pixel data. For 8-bit indexed display data, this bit should be set to a one. |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Bypass | Plane pixel data bypasses the display pipe gamma correction logic | All |
| 1b | Correct | Plane pixel data is gamma corrected in the display pipe gamma correction logic. | All |

## DSPACNTR—Primary A Control Register

29:26 **Source_Pixel_Format**

Project: All

Default Value: 0b

These bits should only be changed after the plane has been disabled. Pixel format of 8-bit indexed uses the pipe palette. Before entering the blender, each source format is converted to 12 bits per pixel.

| Value | Name | Description | Project |
|---|---|---|---|
| 0010b | 8bpp | 8-bpp Indexed | All |
| 0101b | 16-bit BGRX (5:6:5 MSB-R:G:B) | pixel format (XGA compatible). | All |
| 0110b | 32-bit BGRX (8:8:8:8 MSB-X:R:G:B) | pixel format. Ignore alpha | All |
| 1000b | 32-bit RGBX (2:10:10:10 MSB-X:B:G:R) | pixel format. Ignore alpha | All |
| 1010b | 32-bit BGRX (2:10:10:10 MSB-X:R:G:B) | pixel format. Ignore alpha | All |
| 1100b | 64-bit RGBX Floating Point(16:16:16:16 MSB-X:B:G:R)<br>Use of 64bpp format will limit the maximum dot clock to 80% of cdclk | pixel format. Ignore alpha | All |
| 1110b | 32-bit RGBX (8:8:8:8 MSB-X:B:G:R) | pixel format. Ignore alpha | All |
| others | Reserved | Reserved | All |

25 **Reserved** Project: DevSNB Format:

24 **Pipe_Color_Space_Conversion_Enable**

Project: All

Default Value: 0b

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

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

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

15 **180°_Display_Rotation**

Project: All

Default Value: 0b

This mode causes the plane to be rotated 180°. In addition to setting this bit, software must also set the surface address offset to the lower right corner of the unrotated image.

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

## DSPACNTR—Primary A Control Register

**14 — Trickle_Feed_Enable**

Project: All

Default Value: 0b

This bit must always be programmed to '1'.

| 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 |

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

**10 — Tiled_Surface**

Project: All

Default Value: 0b

This bit indicates that the plane surface data is in tiled memory. Only X tiling is supported for display surfaces.

When this bit is set, it affects the hardware interpretation of the DSPATILEOFF, DSPALINOFF, and DSPASURF registers.

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Linear | Plane uses linear memory | All |
| 1b | X-tiled | Plane uses X-tiled memory | All |

**9 — Asynchronous_Surface_Address_Update_Enable**

Project: All

Default Value: 0b

This bit will enable asynchronous updates of the surface address when written by MMIO. The surface address will change with the next 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 surface address updates to this plane are allowed when this bit is enabled.
- Wait for flip done indication in pipe status register before writing the surface address register again with this bit set.

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Sync | Surface Address MMIO writes will update synchronous to start of vertical blank | All |
| 1b | Async | Surface Address MMIO writes will update asynchronously | All |

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

**Primary Plane Source Pixel Format Mapping of Bits to Colors:**

**Note:** For 64-bit Floating Point format, each of the 16 bit color components is 1:5:10 MSB-sign:exponent:fraction

| PRIMARY RGB | 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 | 63:48 | 15:0 | 31:16 | 47:32 |
| 32-bit RGBX 8:8:8 | 31:24 | 7:0 | 15:8 | 23:16 |

## 4.8.2 DSPALINOFF—Primary A Linear Offset Register

### DSPALINOFF—Primary A Linear Offset Register

| | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **70184h-70187h** |
| **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 | **Primary_Linear_Offset** Project: All Format:<br>This register provides the linear panning byte offset into the primary plane. 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. 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 contents of this register are ignored. |

### 4.8.3 DSPASTRIDE—Primary A Stride Register

| DSPASTRIDE—Primary A Stride Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **70188h-7018Bh** |
| **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 DSPASURF |

| Bit | Description |
|---|---|
| 31:16 | **Reserved** Project: All Format: |
| 15:6 | **Primary_Stride** Project: All Format:<br>This is the stride for the primary plane in bytes. When using linear memory, this must be 64 byte aligned. When using tiled memory, this must be 512 byte aligned. This value is used to determine the line to line increment for the display. This register is updated through either a command packet passed through the command stream or writes to this register. The stride is limited to a maximum of 32K bytes for both linear and tiled memory. |
| 5:0 | **Reserved** Project: All Format: |

### 4.8.4 DSPASURF—Primary A Surface Base Address Register

| DSPASURF—Primary A Surface Base Address Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **7019Ch-7019Fh** |
| **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 DSPA registers | |

| DSPASURF—Primary A Surface Base Address Register | |
|---|---|
| **Bit** | **Description** |
| 31:12 | **Primary_Surface_Base_Address**<br>Project: All<br>Address: GraphicsAddress[31:12]<br>This address specifies the surface base address. When the surface is tiled, panning is specified using (x, y) offsets in the DSPATILEOFF register. When the surface is in linear memory, panning is specified using a linear offset in the DSPALINOFF register.<br>This address must be 4K aligned. When performing asynchronous flips and the display surface is in tiled memory, this address must be 256K aligned. This register can be written directly through software or by command packets in the command stream. It represents an offset from the graphics memory aperture base and is mapped to physical pages through the global GTT. |
| 11:3 | **Reserved** Project: All Format: |
| 2 | **Reserved** |
| 0 | **Reserved** Project: All Format: MBZ |

## 4.8.5 DSPATILEOFF—Primary A Tiled Offset Register

| DSPATILEOFF—Primary A Tiled Offset Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **701A4h-701A7h** |
| **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 display surface. The surface base address is specified in the DSPASURF register, and this register is used to describe an offset from that base address. When the surface is in linear memory, the offset is specified in the DSPALINOFF register and the contents of this register are ignored. When the surface is tiled, the start position is specified in this register as an (x, y) offset from the beginning of the surface.

| Bit | Description |
|---|---|
| 31:28 | **Reserved** Project: All Format: MBZ |
| 27:16 | **Primary_Start_Y-Position** Project: All Format:<br>These 12 bits specify the vertical position in lines of the beginning of the active display plane relative to the display surface. When performing 180° rotation, this field specifies the vertical position of the lower right corner relative to the start of the active display plane in the unrotated orientation. |
| 15:12 | **Reserved** Project: All Format: MBZ |

| DSPATILEOFF—Primary A Tiled Offset Register | |
|---|---|
| 11:0 | **Primary_Start_X-Position** Project: All Format:<br>These 12 bits specify the horizontal offset in pixels of the beginning of the active display plane relative to the display surface. When performing 180° rotation, this field specifies the horizontal position of the lower right corner relative to the start of the active display plane in the unrotated orientation. |

## 4.8.6 DSPASURFLIVE—Primary A Live Surface Base Address

| DSPASURFLIVE—Primary A Live Surface Base Address | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **701ACh-701AFh** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | Read Only |
| **Size (in bits):** | 32 |
| **Trusted Type:** | 1 |
| **Bit** | **Description** |
| 31:0 | **Primary_Live_Surface_Base_Address**<br>Project: All<br>Address: GraphicsAddress[31:0]<br>This gives the live value of the surface base address as being currently used for the plane. |

## 4.9 Primary B Plane Control

The DSPBCNTR and DSPBSTRIDE active registers will be updated on the vertical blank or when pipe is disabled, after the DSPBSURF trigger register is written, or when the primary B is not yet enabled – thus providing an atomic update of the primary B control, stride, and base address registers.

### 4.9.1 DSPBCNTR—Primary B Control Register

**DSPBCNTR—Primary B Control Register**

| | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **71180h-71183h** |
| **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 DSPBSURF |

Primary B Plane is connected to pipe B only

| Bit | Description |
|---|---|
| 31 | **Primary_Plane_Enable** Project: All Format: Enable<br>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. The display pipe must be enabled to enable the plane. When in Self Refresh Big FIFO mode, write to this register to enable the plane will be internally buffered and delayed while Big FIFO mode is exiting. |
| 30 | **Gamma_Enable**<br>Project: All<br>Default Value: 0b<br>This bit should only be changed after the plane has been disabled. It controls the bypassing of the display pipe gamma unit for the plane pixel data. For 8-bit indexed display data, this bit should be set to a one. |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Bypass | Plane pixel data bypasses the display pipe gamma correction logic | All |
| 1b | Correct | Plane pixel data is gamma corrected in the display pipe gamma correction logic. | All |

## DSPBCNTR—Primary B Control Register

29:26 **Source_Pixel_Format**

Project: All

Default Value: 0b

These bits should only be changed after the plane has been disabled. Pixel format of 8-bit indexed uses the pipe palette. Before entering the blender, each source format is converted to 12 bits per pixel.

| Value | Name | Description | Project |
|---|---|---|---|
| 0010b | 8bpp | 8-bpp Indexed | All |
| 0101b | 16-bit BGRX (5:6:5 MSB-R:G:B) | pixel format (XGA compatible). | All |
| 0110b | 32-bit BGRX (8:8:8:8 MSB-X:R:G:B) | pixel format. Ignore alpha | All |
| 1000b | 32-bit RGBX (2:10:10:10 MSB-X:B:G:R) | pixel format. Ignore alpha | All |
| 1010b | 32-bit BGRX (2:10:10:10 MSB-X:R:G:B) | pixel format. Ignore alpha | All |
| 1100b | 64-bit RGBX Floating Point(16:16:16:16 MSB-X:B:G:R)<br>Use of 64bpp format will limit the maximum dot clock to 80% of cdclk. | pixel format. Ignore alpha | All |
| 1110b | 32-bit RGBX (8:8:8:8 MSB-X:B:G:R) | pixel format. Ignore alpha | All |
| others | Reserved | Reserved | All |

25 **Reserved** Project: DevSNB

24 **Pipe_Color_Space_Conversion_Enable**

Project: All

Default Value: 0b

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

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

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

15 **180°_Display_Rotation**

Project: All

Default Value: 0b

This mode causes the plane to be rotated 180°. In addition to setting this bit, software must also set the surface address offset to the lower right corner of the unrotated image.

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

## DSPBCNTR—Primary B Control Register

**14 — Trickle_Feed_Enable**

Project: All

Default Value: 0b

This bit must always be programmed to '1'.

| 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 — Reserved**

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

**10 — Tiled_Surface**

Project: All

Default Value: 0b

This bit indicates that the plane surface data is in tiled memory. Only X tiling is supported for display surfaces.

When this bit is set, it affects the hardware interpretation of the DSPBTILEOFF, DSPBLINOFF, and DSPBSURF registers.

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Linear | Plane uses linear memory | All |
| 1b | X-tiled | Plane uses X-tiled memory | All |

**9 — Asynchronous_Surface_Address_Update_Enable**

Project: All

Default Value: 0b

This bit will enable asynchronous updates of the surface address when written by MMIO. The surface address will change with the next 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 surface address updates to this plane are allowed when this bit is enabled.
- Wait for flip done indication in pipe status register before writing the surface address register again with this bit set.

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Sync | Surface Address MMIO writes will update synchronous to start of vertical blank | All |
| 1b | Async | Surface Address MMIO writes will update asynchronously | All |

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

**See DSPACNTR - Primary Plane Source Pixel Format Mapping of Bits to Colors**

## 4.9.2 DSPBLINOFF—Primary B Linear Offset Register

| DSPBLINOFF—Primary B Linear Offset Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **71184h-71187h** |
| **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 | **Primary_Linear_Offset** Project: All<br>This register provides the linear panning byte offset into the primary plane. 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. 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 contents of this register are ignored. |

## 4.9.3 DSPBSTRIDE—Primary B Stride Register

| DSPBSTRIDE—Primary B Stride Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **71188h-7118Bh** |
| **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 DSPBSURF |

| Bit | Description |
|---|---|
| 31:16 | **Reserved** Project: All Format: MBZ |
| 15:6 | **Primary_Stride** Project: All<br>This is the stride for the primary plane in bytes. When using linear memory, this must be 64 byte aligned. When using tiled memory, this must be 512 byte aligned. This value is used to determine the line to line increment for the display. This register is updated through either a command packet passed through the command stream or writes to this register. The stride is limited to a maximum of 32K bytes for both linear and tiled memory. |
| 5:0 | **Reserved** Project: All Format: MBZ |

### 4.9.4 DSPBSURF—Primary B Surface Base Address Register

| DSPBSURF—Primary B Surface Base Address Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **7119Ch-7119Fh** |
| **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 DSPB registers

| Bit | Description |
|---|---|
| 31:12 | **Primary_Surface_Base_Address**<br>Project: All<br>Address: GraphicsAddress[31:12]<br>This address specifies the surface base address. When the surface is tiled, panning is specified using (x, y) offsets in the DSPBTILEOFF register. When the surface is in linear memory, panning is specified using a linear offset in the DSPBLINOFF register.<br>This address must be 4K aligned. When performing asynchronous flips and the display surface is in tiled memory, this address must be 256K aligned. This register can be written directly through software or by command packets in the command stream. It represents an offset from the graphics memory aperture base and is mapped to physical pages through the global GTT. |
| 11:3 | **Reserved** Project: All Format: |
| 2 | **Reserved** |
| 0 | **Reserved** Project: All Format: MBZ |

### 4.9.5 DSPBTILEOFF—Primary B Tiled Offset Register

Address Offset: 711A4h

Default: 00000000h

Normal Access: Read/Write
Double Buffer Update Point: Start of vertical blank or pipe disabled

Size: 32 bits

This register specifies the panning for the display surface. The surface base address is specified in the DSPBSURF register, and this register is used to describe an offset from that base address. When the surface is in linear memory, the offset is specified in the DSPBLINOFF register and the contents of this register are ignored. When the surface is tiled, the start position is specified in this register as an (x, y) offset from the beginning of the surface.

| Bit | Descriptions |
|---|---|
| 31:28 | **Reserved:** Write as zero |
| 27:16 | **Primary Start Y-Position:** These 12 bits specify the vertical position in lines of the beginning of the active display plane relative to the display surface. When performing 180° rotation, this field specifies the vertical position of the lower right corner relative to the start of the active display plane in the unrotated orientation. |
| 15:12 | **Reserved:** Write as zero |
| 11:0 | **Primary Start X-Position:** These 12 bits specify the horizontal offset in pixels of the beginning of the active display plane relative to the display surface. When performing 180° rotation, this field specifies the horizontal position of the lower right corner relative to the start of the active display plane in the unrotated orientation. |

### 4.9.6 DSPBSURFLIVE—Primary B Live Surface Base Address

| DSPBSURFLIVE—Primary B Live Surface Base Address | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **711ACh-711AFh** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | Read Only |
| **Size (in bits):** | 32 |

| Bit | Description |
|---|---|
| 31:0 | **Primary_Live_Surface_Base_Address**<br>Project: All<br>Address: GraphicsAddress[31:0]<br>This gives the live value of the surface base address as being currently used for the plane. |

## 4.10 Video Sprite A Control

Two video sprites are provided for the display of video content. These sprite planes provide windowing and keying functions as well as gamma and color space conversion from YUV to RGB. Each sprite plane is attached to only one of the pipes, Video Sprite A to pipe A and Video Sprite B to pipe B. Apart from the pipe assignments, the functionality is identical.

The DVSACNTR, DVSASTRIDE, DVSAPOS, DVSASIZE, and DVSASCALE active registers will be updated on the vertical blank or when pipe is disabled, after the DVSASURF trigger register is written, or when the sprite A is not yet enabled – thus providing an atomic update of the video sprite A control, stride, position, size, scale, and base address registers.

## 4.10.1 DVSACNTR—Video Sprite A Control Register

### DVSACNTR—Video Sprite A Control Register

| | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **72180h-72183h** |
| **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 DVSASURF |

Video Sprite A Plane is connected to pipe A only.

| Bit | Description |
|---|---|
| 31 | **Sprite_Enable** Project: All Format: Enable<br>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. The display pipe must be enabled to enable the plane. When in Self Refresh Big FIFO mode, write to this register to enable the plane will be internally buffered and delayed while Big FIFO mode is exiting. |
| 30 | **Gamma_Enable**<br>Project: All<br>Default Value: 0b<br>There are two gamma adjustments possible in the video sprite data path. This bit controls the gamma correction in the display pipe not the gamma control in this plane. It affects only the pixel data from this display plane. Gamma correction logic that is contained in the video sprite is disabled by loading the default values into those registers. |
| 29 | **Reserved** Project: All Format: MBZ |
| 28 | **YUV_Bypass_Excess-512_Format_Conversion**<br>Project: All<br>Default Value: 0b |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Disable | Plane pixel data bypasses the display pipe gamma correction logic | All |
| 1b | Enable | Plane pixel data is gamma corrected in the pipe gamma correction logic | All |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Disable | Disable excess-512 conversion | All |
| 1b | Enable | Enable excess-512 conversion | All |

## DVSACNTR—Video Sprite A Control Register

27

**Range_Correction_Disable**

Project: All

Default Value: 0b

Setting this bit disables the YUV range correction logic. Normally the range compressed YUV is expanded to full range RGB, setting this bit will generate range compressed RGB. This bit should also be used if full range YUV source material is used. This bit has no effect on RGB source formats.

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Enable | Range correction enabled | All |
| 1b | Disable | No range correction | All |

26:25

**Source_Pixel_Format**

Project: DevSNB

Default Value: 0b

This field selects the pixel format for the sprite. Before entering the blender, each source format is converted to 12 bits per pixel.

| Value | Name | Description | Project |
|---|---|---|---|
| 00b | YUV 4:2:2 | YUV 4:2:2 packed pixel format (byte order programmed separately) | All |
| 01b | RGB 32-bit 2:10:10:10 | RGB 32-bit 2:10:10:10 pixel format (color order programmed separately). Ignore alpha. | All |
| 10b | RGB 32-bit 8:8:8:8 | RGB 32-bit 8:8:8:8 pixel format (color order programmed separately). Ignore alpha. | All |
| 11b | RGB 64-bit 16:16:16:16<br>Use of 64bpp format will limit the maximum dot clock to 80% of cdclk. | RGB 64-bit 16:16:16:16 floating point pixel format (color order programmed separately). Ignore alpha. | All |

24

**Pipe_Color_Space_Conversion_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. CSC mode in the pipe CSC registers must be set to match the format of the plane pixel data after the color conversion logic within the sprite plane.

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

23

**Reserved** Project: All Format: MBZ

## DVSACNTR—Video Sprite A Control Register

**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: DevSNB Format:

**20 — RGB_Color_Order**

Project: DevSNB

Default Value: 0b

This field is used to select the color order when using RGB data formats. 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 |

**20 — Reserved** Project: DevILK Format:

**19 — Color_Conversion_Disabled**

Project: All

Default Value: 0b

This bit enables or disables the color conversion logic internal to the sprite. Color conversion is intended to be used with the formats that support YUV format. This bit is ignored when using RGB source formats.

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Enable | Pixel data is sent through the sprite color conversion logic (only applies to YUV formats) | All |
| 1b | Disable | Pixel data is not sent through the sprite YUV->RGB color conversion logic. | All |

## DVSACNTR—Video Sprite A Control Register

18 **YUV_Format**

Project: All

Default Value: 0b

This bit specifies the source YUV format for the YUV to RGB color 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_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 |

15 **180°_Display_Rotation**

Project: All

Default Value: 0b

This mode causes the plane to be rotated 180°. In addition to setting this bit, software must also set the surface address offset to the lower right corner of the unrotated and unscaled image and calculate the x, y offset as relative to the lower right corner.

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

14 **Trickle-Feed_Enable**

Project: DevSNB

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 |

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

## DVSACNTR—Video Sprite A Control Register

**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 DVSASTRIDE register. Only X tiling is supported for display surfaces.

When this bit is set, it affects the hardware interpretation of the DVSASTART and DVSASURFADDR registers.

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Linear | Linear memory | All |
| 1b | Tiled | Tiled memory | All |

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

**2 — Sprite_Destination_Key**

Project: All

Default Value: 0b

This bit enables the destination key function. If the pixel for the primary plane on this pipe matches the key value in DVSAKEYVAL the sprite pixel is used, otherwise the primary plane pixel is passed through the blender unmodified. 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 source pixel format mapping

| SPRITE YUV | 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 |

## 4.10.2 DVSALINOFF—Video Sprite A Linear Offset Register

| DVSALINOFF—Video Sprite A Linear Offset Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **72184h-72187h** |
| **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 | **Sprite_Linear_Offset** Project: All Format:<br>This register provides the linear panning byte offset into the sprite plane. 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 unrotated RGB formats and even pixel aligned for unrotated YUV 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 contents of this register are ignored. |

## 4.10.3 DVSASTRIDE—Video Sprite A Stride Register

| DVSASTRIDE—Video Sprite A Stride Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **72188h-7218Bh** |
| **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 DVSASURF |

| Bit | Description |
|---|---|
| 31:15 | **Reserved** Project: All Format: |
| 14:6 | **Sprite_Stride** Project: All Format:<br>This is the stride for the sprite in bytes. When using linear memory, this must be 64 byte aligned. When using tiled memory, this must be 512 byte aligned. This register is updated through either a command packet passed through the command stream or writes to this register. The stride is limited to a maximum of 16K bytes when sprite scaling is not enabled, 4K bytes when sprite scaling is enabled, for both linear and tiled memory. |
| 5:0 | **Reserved** Project: All Format: |

## 4.10.4 DVSAPOS—Video Sprite A Position Register

| DVSAPOS—Video Sprite A Position Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **7218Ch-7218Fh** |
| **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 DVSASURF, DVSAPOS write does not disarm |

This register specifies the position of the sprite. Software must take care that the sprite does not extend out of the display active area. ie. Xposition + Xsize =< Xsrcsize

| Bit | Description |
|---|---|
| 31:28 | **Reserved** Project: All Format: MBZ |
| 27:16 | **Sprite_Y-Position** Project: All Format:<br>These 12 bits specify the vertical position in lines of the sprite (upper left corner) relative to the beginning of the active video area. When performing 180° rotation, this field specifies the vertical position of the lower right corner relative to the end of the active video area in the unrotated orientation. The defined sprite rectangle must always be completely contained within the displayable area of the screen image. |
| 15:12 | **Reserved** Project: All Format: MBZ |
| 11:0 | **Sprite_X-Position** Project: All Format:<br>These 12 bits specify the horizontal position in pixels of the sprite (upper left corner) relative the beginning of the active video area. When performing 180° rotation, this field specifies the horizontal position of the original lower right corner relative to the original end of the active video area in the unrotated orientation. The defined sprite rectangle must always be completely contained within the displayable area of the screen image. |

## 4.10.5 DVSASIZE—Video Sprite A Size Register

| DVSASIZE—Video Sprite A Size Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **72190h-72193h** |
| **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 DVSASURF |

This register specifies the size of the sprite. Software must take care that the sprite does not extend out of the display active area. ie. Xposition + Xsize =< Xsrcsize

| Bit | Description |
|---|---|
| 31:28 | **Reserved** Project: All Format: MBZ |
| 27:16 | **Sprite_Height** Project: All Format:<br>This register field is used to specify the height of the sprite in lines. The value in the register is the height minus one. The defined sprite rectangle must always be completely contained within the displayable area of the screen image. |
| 15:12 | **Reserved** Project: All Format: MBZ |
| 11:0 | **Sprite_Width** Project: All Format:<br>This register field is used to specify the width of the sprite in pixels. This does not have to be the same as the stride but should be less than or equal to the stride in pixels. The value in the register is the width minus one. The defined sprite rectangle must always be completely contained within the displayable area of the screen image.<br>The sprite width (actual width, not the width minus one value) is limited to even values when YUV source pixel format is used, or Pixel Multiply is set to Line/Pixel doubling or Pixel doubling. |

## 4.10.6 DVSAKEYVAL—Video Sprite A Color Key Value Register

| DVSAKEYVAL—Video Sprite A Color Key Value Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **72194h-72197h** |
| **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 key color to be used with the mask bits to determine if the sprite source data matches the key. This register will only have an effect when the sprite color key is enabled. In source key mode this value is the minimum value for the range compare. In destination key mode this value is the compare value.

| Bit | Description |
|---|---|
| 31:24 | **Reserved** Project: All Format: MBZ |
| 23:16 | **V_Source_Key_Min_Value/R_Source/Dest_Key_Value** Project: All Format:<br>Specifies the color key (minimum) value for the sprite V channel source key or the Red channel source or destination key compare value. |
| 15:8 | **Y_Source_Key_Min_Value/G_Source/Dest_Key_Value** Project: All Format:<br>Specifies the color key (minimum) value for the sprite Y channel source key or the Green channel source or destination key compare value. |
| 7:0 | **U_Source_Key_Min_Value/B_Source/Dest_Key_Value** Project: All Format:<br>Specifies the color key (minimum) value for the sprite U channel source key or the Blue channel source or destination key compare value. |

## 4.10.7 DVSAKEYMSK—Video Sprite A Color Key Mask Register

### DVSAKEYMSK—Video Sprite A Color Key Mask Register

| | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **72198h-7219Bh** |
| **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 range checking on.

For destination key this register specifies the key mask to be used with the color value bits to determine if the display source data matches the key when enabled. 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_Channel_Source_Key_Enable** Project: All Format:<br>Specifies the source color key enable for the V/Red channel |
| 25 | **Y/G_Channel_Source_Key_Enable** Project: All Format:<br>Specifies the source color key enable for the Y/Green channel |
| 24 | **U/B_Channel_Source_Key_Enable** Project: All Format:<br>Specifies the source color key enable for the U/Blue channel |
| 23:16 | **R_mask_Dest_Key_Value** Project: All Format:<br>Specifies the destination color key mask for the sprite R channel |
| 15:8 | **G_mask_Dest_Key_Value** Project: All Format:<br>Specifies the destination color key mask for the sprite G channel |
| 7:0 | **B_mask_Dest_Key_Value** Project: All Format:<br>Specifies the destination color key mask for the sprite B channel |

## 4.10.8 DVSASURF—Video Sprite A Surface Address Register

| DVSASURF—Video Sprite A Surface Address Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **7219Ch-7219Fh** |
| **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 DVSA registers** | |

| Bit | Description |
|---|---|
| 31:12 | **Sprite_Surface_Base_Address**<br>Project: All<br>Address: Graphicsdress[31:12]<br>This address specifies the surface base address. When the surface is tiled, panning is specified using (x, y) offsets in the DVSATILEOFF register. When the surface is in linear memory, panning is specified using a linear offset in the DVSALINOFF register.<br>This address must be 4K aligned. It represents an offset from the graphics memory aperture base and is mapped to physical pages through the global GTT. The value in this register is updated through the command streamer with synchronous flips. |
| 11:0 | **Reserved** Project: All Format: |

### 4.10.9 DVSAKEYMAXVAL—Video Sprite A Color Key Max Value Register

| DVSAKEYMAXVAL—Video Sprite A Color Key Max Value Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **721A0h-721A3h** |
| **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 key color to be used with the mask bits to determine if the sprite source data matches the key. This register will only have an effect when the sprite source color key is enabled

| Bit | Description |
|---|---|
| 31:24 | **Reserved** Project: All Format: MBZ |
| 23:16 | **V_Key_Max_Value** Project: All Format:<br>Specifies the color key value for the sprite V channel |
| 15:8 | **Y_Key_Max_Value** Project: All Format:<br>Specifies the color key value for the sprite Y channel |
| 7:0 | **U_Key_Max_Value** Project: All Format:<br>Specifies the color key value for the sprite U channel |

## 4.10.10 DVSATILEOFF—Video Sprite A Tiled Offset Register

### DVSATILEOFF—Video Sprite A Tiled Offset Register

| | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **721A4h-721A7h** |
| **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 sprite surface in tiled memory. The surface base address is specified in the DVSASURFADDR register, and this register is used to describe an offset from that base address. When the surface is in linear memory, the offset is specified in the DVSALINOFF register and the contents of this register are ignored. When the surface is tiled, the start position is specified in this register as an (x, y) offset from the beginning of the surface.

| Bit | Description |
|---|---|
| 31:28 | **Reserved** Project: All Format: MBZ |
| 27:16 | **Sprite_Start_Y-Position** Project: All Format:<br>These 12 bits specify the vertical position in lines of the beginning of the active display plane relative to the display surface. When performing 180° rotation, this field specifies the vertical position of the lower right corner relative to the start of the active display plane in the unrotated orientation. |
| 15:12 | **Reserved** Project: All Format: MBZ |
| 11:0 | **Sprite_Start_X-Position** Project: All Format:<br>These 12 bits specify the horizontal offset in pixels of the beginning of the active display plane relative to the display surface. The offset must be even pixel aligned for unrotated YUV formats. When performing 180° rotation, this field specifies the horizontal position of the lower right corner relative to the start of the active display plane in the unrotated orientation. |

## 4.10.11 DVSASURFLIVE—Video Sprite A Live Surface Base Address Register

| DVSASURFLIVE—Video Sprite A Live Surface Base Address Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **721ACh-721AFh** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | Read Only |
| **Size (in bits):** | 32 |

| Bit | Description |
|---|---|
| 31:0 | **Sprite_Surface_Base_Address**<br>Project: All<br>Address: GraphicsAddress[31:0]<br>This gives the live value of the surface base address as being currently used for the plane. |

## 4.10.12DVSASCALE—Video Sprite A Scaler Control

### DVSASCALE—Video Sprite A Scaler Control

| | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **72204h-72207h** |
| **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 DVSASURF |

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

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. Scaling can not be used with the sprite 64bpp source pixel format. Source and destination sizes must be 3x3 (3x6 when interlacing) or greater when scaling is enabled.

Horizontal downscaling limits the maximum dot clock allowed as percent of cdclk. Rules to calculate the allowed dot clock:

Start with maximum dot clock 90% of cdclk. (There is a separate requirement that planes using 64bpp formats can not be enabled with dot clock >80% of cdclk. Subtract 10% more per horizontal decimation step (decimation steps at 2x, 4x, 8x, and 16x downscale).

Subtract 10% more if sprite is using the RGB data format.

Subtract 10% more if sprite scaling is enabled on the other pipe.

Then divide that by horizontal downscale amount within each decimation step.

The result is the maximum allowed dot clock as percent of cdclk frequency.

**Example:**

| Scale factor | Decimation amount | YUV single pipe dot clock % | YUV dual pipe dot clock % | RGB single pipe dot clock % | RGB dual pipe dot clock % | Comment |
|---|---|---|---|---|---|---|
| 1 | 1 | 90 | 80 | 80 | 70 | No scaling |
| 1.5 | 1 | 60 | 53 | 53 | 46 | |
| 1.99 | 1 | 45 | 40 | 40 | 35 | Max downscale before decimation starts |
| 2 | 2 | 80 | 70 | 70 | 60 | |
| 3 | 2 | 53 | 46 | 46 | 40 | |
| 3.99 | 2 | 40 | 35 | 35 | 30 | |
| 4 | 4 | 70 | 60 | 60 | 50 | |

## DVSASCALE—Video Sprite A Scaler Control

| | | | | | | |
|---|---|---|---|---|---|---|
| 6 | 4 | 46 | 40 | 40 | 33 | |
| 7.99 | 4 | 35 | 30 | 30 | 25 | |
| 8 | 8 | 60 | 50 | 50 | 40 | |
| 12 | 8 | 40 | 33 | 33 | 26 | |
| 15.99 | 8 | 30 | 25 | 25 | 20 | Worst case dot clock |
| 16 | 16 | 50 | 40 | 40 | 30 | Max downscaling allowed |

| Bit | Description |
|---|---|
| 31 | **Scaling_Enable** Project: All Format: Enable<br>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. |
| 30:29 | **Filter_Control**<br>Project: All<br>Default Value: 0b<br>Filter selection |

| Value | Name | Description | Project |
|---|---|---|---|
| 00b | Medium | Medium Filtering | All |
| 01b | Enhancing | Edge Enhancing Filtering | All |
| 10b | Softening | Edge Softening Filtering | All |
| 11b | Reserved | Reserved | All |

| Bit | Description |
|---|---|
| 28 | **Even/Odd_Field_Offset**<br>Project: All<br>Default Value: 0b<br>Select the vertical offset of the filtered data. Software is responsible for updating this to match the surface data. |

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | 0 | Vertical initial phase of 0 | All |
| 1b | 0.5 | Vertical initial phase of 0.5 | All |

| DVSASCALE—Video Sprite A Scaler Control | |
|---|---|
| 27 | **Even/Odd_Field_Enable**<br>Project: All<br>Default Value: 0b<br>Enable adjustment of the vertical offset of the filtered data.<br>**Value** / **Name** / **Description** / **Project**<br>0b / Disable / Off (Vertical initial phase is 1/2 the scale factor) / All<br>1b / Enable / On (Vertical initial phase is selected by the Even/Off Field Offset bit) / All |
| 26:16 | **Source_Width** Project: All Format:<br>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 source pixel format is used. |
| 15:11 | **Reserved** Project: All Format: MBZ |
| 10:0 | **Source_Height** Project: All Format:<br>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.<br>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. |

## 4.10.13 DVSAGAMC—Video Sprite A Gamma Correction Registers

| DVSGAMC Reference Point | | |
|---|---|---|
| **Project:** | All | |
| **Bit** | **Description** | |
| 31:30 | **Reserved** | Project: All Format: |
| 29:20 | **Red Gamma Reference Point** | Project: All Format: |
| 19:10 | **Green Gamma Reference Point** | Project: All Format: |
| 9:0 | **Blue Gamma Reference Point** | Project: All Format: |

| DVSGAMC Max Reference Point | | |
|---|---|---|
| **Project:** | All | |
| **Bit** | **Description** | |
| 31:11 | **Reserved** Project: All | Format: |
| 10:0 | **Final Gamma Max Reference Point** | Project: All Format: |

## DVSAGAMC—Video Sprite A Gamma Correction Registers

| | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **72300h-7234Bh** |
| **Project:** | All |
| **Default Value:** | 00000000h; 04010040h; 08020080h; 0C0300C0h; 10040100h; 14050140h; 18060180h; 1C0701C0h; 20080200h; 24090240h; 280A0280h; 2C0B02C0h; 300C0300h; 340D0340h; 380E0380h; 3C0F03C0h; 00000400h; 00000400h; 00000400h; |
| **Access:** | R/W |
| **Size (in bits):** | 19x32 |

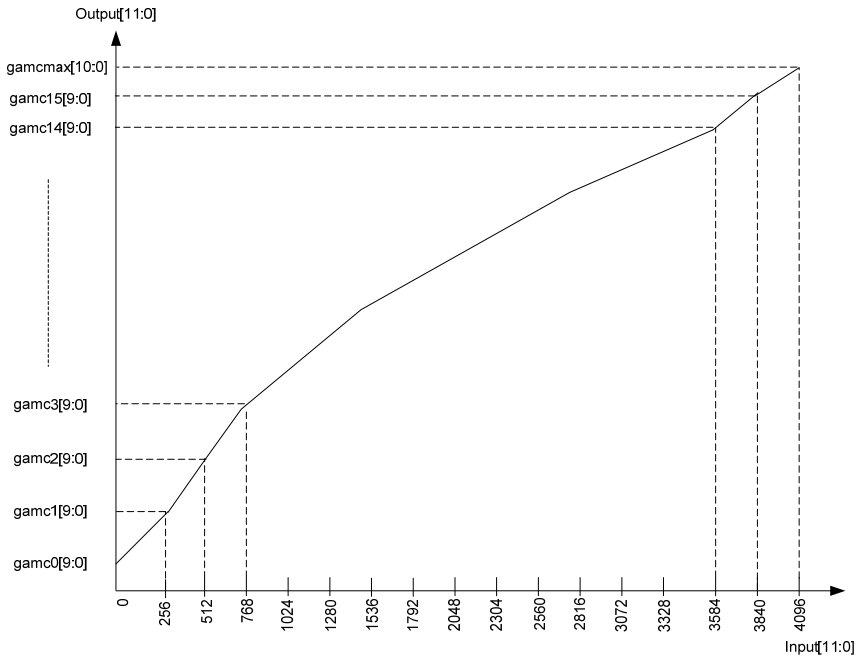
These registers are used to determine the characteristics of the gamma correction for the sprite pixel data *pre-blending*. Additional gamma correction can be done in the display pipe gamma if desired.

The gamma correction curve is represented by specifying a set of reference points spaced equally along the desired curve. Red, Green, and Blue each have 17 reference points. The first 16 reference points are 10 bit values with Red, Green, and Blue sharing a single register for each point. The final max reference point is an 11 bit value with separate registers for Red, Green, and Blue. The curve must be flat or increasing, never decreasing.

During operation the appropriate reference point pairs (adjacent) are selected for each color, and the output is interpolated between these two reference point values.

The gamma correction registers are not double-buffered. They should only be updated when the sprite is off, otherwise, screen artifacts may show.

To pass sprite pixel data through gamma correction unchanged, program the gamma reference points to the default linear ramp values. When the output from sprite is set in YUV format by programming CSC bypass, the sprite gamma correction will be bypassed.




**Programming of the Piecewise-linear Estimation of Gamma Correction Curve**

| DVSAGAMC—Video Sprite A Gamma Correction Registers | | | | | | |
|---|---|---|---|---|---|---|
| **DWord** | **Bit** | **Description** | | | | |
| 0 | 31:0 | **GAMC0** | Project: | All | Format: | DVSGAMC Reference Point |
| 1 | 31:0 | **GAMC1** | Project: | All | Format: | DVSGAMC Reference Point |
| 2 | 31:0 | **GAMC2** | Project: | All | Format: | DVSGAMC Reference Point |
| 3 | 31:0 | **GAMC3** | Project: | All | Format: | DVSGAMC Reference Point |
| 4 | 31:0 | **GAMC4** | Project: | All | Format: | DVSGAMC Reference Point |
| 5 | 31:0 | **GAMC5** | Project: | All | Format: | DVSGAMC Reference Point |
| 6 | 31:0 | **GAMC6** | Project: | All | Format: | DVSGAMC Reference Point |
| 7 | 31:0 | **GAMC7** | Project: | All | Format: | DVSGAMC Reference Point |
| 8 | 31:0 | **GAMC8** | Project: | All | Format: | DVSGAMC Reference Point |
| 9 | 31:0 | **GAMC9** | Project: | All | Format: | DVSGAMC Reference Point |
| 10 | 31:0 | **GAMC10** | Project: | All | Format: | DVSGAMC Reference Point |
| 11 | 31:0 | **GAMC11** | Project: | All | Format: | DVSGAMC Reference Point |
| 12 | 31:0 | **GAMC12** | Project: | All | Format: | DVSGAMC Reference Point |
| 13 | 31:0 | **GAMC13** | Project: | All | Format: | DVSGAMC Reference Point |
| 14 | 31:0 | **GAMC14** | Project: | All | Format: | DVSGAMC Reference Point |
| 15 | 31:0 | **GAMC15** | Project: | All | Format: | DVSGAMC Reference Point |
| 16 | 31:0 | **GAMCmaxR** | Project: | All | Format: | DVSGAMC Max Reference Point |
| 17 | 31:0 | **GAMCmaxG** | Project: | All | Format: | DVSGAMC Max Reference Point |
| 18 | 31:0 | **GAMCmaxB** | Project: | All | Format: | DVSGAMC Max Reference Point |

## 4.11 Video Sprite B Control

The DVSBCNTR, DVSBSTRIDE, DVSBPOS, DVSBSIZE, and DVSBSCALE active registers will be updated on the vertical blank or when pipe is disabled, after the DVSBSURF trigger register is written, or when the sprite B is not yet enabled – thus providing an atomic update of the video sprite B control, stride, position, size, scale, and base address registers.

## 4.11.1 DVSBCNTR—Video Sprite B Control Register

### DVSBCNTR—Video Sprite B Control Register

| | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **73180h-73183h** |
| **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 DVSBSURF |

Video Sprite B Plane is connected to pipe B only.

| Bit | Description |
|---|---|
| 31 | **Sprite_Enable** Project: All Format: Enable<br>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. The display pipe must be enabled to enable the plane. When in Self Refresh Big FIFO mode, write to this register to enable the plane will be internally buffered and delayed while Big FIFO mode is exiting. |
| 30 | **Gamma_Enable**<br>Project: All<br>Default Value: 0b<br>There are two gamma adjustments possible in the video sprite data path. This bit controls the gamma correction in the display pipe not the gamma control in this plane. It affects only the pixel data from this display plane. Gamma correction logic that is contained in the video sprite is disabled by loading the default values into those registers. |
| 29 | **Reserved** Project: All Format: MBZ |
| 28 | **YUV_Bypass_Excess-512_Format_Conversion**<br>Project: All<br>Default Value: 0b |

Bit 30 — Gamma_Enable

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Disable | Plane pixel data bypasses the display pipe gamma correction logic | All |
| 1b | Enable | Plane pixel data is gamma corrected in the pipe gamma correction logic | All |

Bit 28 — YUV_Bypass_Excess-512_Format_Conversion

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Disable | Disable excess-512 conversion | All |
| 1b | Enable | Enable excess-512 conversion | All |

## DVSBCNTR—Video Sprite B Control Register

27

**Range_Correction_Disable**

Project: All

Default Value: 0b

Setting this bit disables the YUV range correction logic. Normally the range compressed YUV is expanded to full range RGB, setting this bit will generate range compressed RGB. This bit should also be used if full range YUV source material is used. This bit has no effect on RGB source formats.

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Enable | Range correction enabled | All |
| 1b | Disable | No range correction | All |

26:25

**Source_Pixel_Format**

Project: DevSNB

Default Value: 0b

This field selects the pixel format for the sprite. Before entering the blender, each source format is converted to 12 bits per pixel.

| Value | Name | Description | Project |
|---|---|---|---|
| 00b | YUV 4:2:2 | YUV 4:2:2 packed pixel format (byte order programmed separately) | All |
| 01b | RGB 32-bit 2:10:10:10 | RGB 32-bit 2:10:10:10 pixel format (color order programmed separately). Ignore alpha. | All |
| 10b | RGB 32-bit 8:8:8:8 | RGB 32-bit 8:8:8:8 pixel format (color order programmed separately). Ignore alpha. | All |
| 11b | RGB 64-bit 16:16:16:16<br>Use of 64bpp format will limit the maximum dot clock to 80% of cdclk. | RGB 64-bit 16:16:16:16 floating point pixel format (color order programmed separately). Ignore alpha. | All |

24

**Pipe_Color_Space_Conversion_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. CSC mode in the pipe CSC registers must be set to match the format of the plane pixel data after the color conversion logic within the sprite plane.

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

23

**Reserved** Project: All Format: MBZ

## DVSBCNTR—Video Sprite B Control Register

**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 |

**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 | | Sprite source key is disabled | All |
| 1b | | Sprite source key is enabled | All |

**21** **Reserved** Project: DevSNB

**20** **RGB_Color_Order**

Project: DevSNB

Default Value: 0b

This field is used to select the color order when using RGB data formats. 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 |

**20** **Reserved** Project: DevILK Format:

## DVSBCNTR—Video Sprite B Control Register

**19 — Color_Conversion_Disabled**

Project: All

Default Value: 0b

This bit enables or disables the color conversion logic internal to the sprite. Color conversion is intended to be used with the formats that support YUV format. This bit is ignored when using RGB source formats.

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Enable | Pixel data is sent through the sprite color conversion logic (only applies to YUV formats) | All |
| 1b | Disable | Pixel data is not sent through the sprite YUV->RGB color conversion logic. | All |

**18 — YUV_Format**

Project: All

Default Value: 0b

This bit specifies the source YUV format for the YUV to RGB color 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_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 |

## DVSBCNTR—Video Sprite B Control Register

**15** — **180°_Display_Rotation**

Project: All

Default Value: 0b

This mode causes the plane to be rotated 180°. In addition to setting this bit, software must also set the surface address offset to the lower right corner of the unrotated and unscaled image and calculate the x, y offset as relative to the lower right corner.

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

**14** — **Trickle-Feed_Enable**

Project: DevSNB

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 |

**13:11** — **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 DVSBSTRIDE register. Only X tiling is supported for display surfaces.

When this bit is set, it affects the hardware interpretation of the DVSBSTART and DVSBSURFADDR registers.

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Linear | Linear memory | All |
| 1b | Tiled | Tiled memory | All |

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

| DVSBCNTR—Video Sprite B Control Register | |
|---|---|
| 2 | **Sprite_Destination_Key**<br>Project: All<br>Default Value: 0b<br>This bit enables the destination key function. If the pixel for the primary plane on this pipe matches the key value in DVSBKEYVAL the sprite pixel is used, otherwise the primary plane pixel is passed through the blender unmodified. 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 |

**See DVSACNTR - Sprite Source Pixel Format Mapping of Bits to Colors**

## 4.11.2 DVSBLINOFF—Video Sprite B Linear Offset Register

| DVSBLINOFF—Video Sprite B Linear Offset Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **73184h-73187h** |
| **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 | **Sprite_Linear_Offset** Project: All Format:<br>This register provides the linear panning byte offset into the sprite plane. 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 unrotated RGB formats and even pixel aligned for unrotated YUV 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 contents of this register are ignored. |

### 4.11.3 DVSBSTRIDE—Video Sprite B Stride Register

| DVSBSTRIDE—Video Sprite B Stride Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **73188h-7318Bh** |
| **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 DVSBSURF |

| Bit | Description |
|---|---|
| 31:15 | **Reserved** Project: All Format: |
| 14:6 | **Sprite_Stride** Project: All Format:<br>This is the stride for the sprite in bytes. When using linear memory, this must be 64 byte aligned. When using tiled memory, this must be 512 byte aligned. This register is updated through either a command packet passed through the command stream or writes to this register. The stride is limited to a maximum of 16K bytes when sprite scaling is not enabled, 4K bytes when sprite scaling is enabled, for both linear and tiled memory. |
| 5:0 | **Reserved** Project: All Format: |

### 4.11.4 DVSBPOS—Video Sprite B Position Register

| DVSBPOS—Video Sprite B Position Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **7318Ch-7318Fh** |
| **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 DVSBSURF, DVSAPOS write does not disarm |

This register specifies the position of the sprite. Software must take care that the sprite does not extend out of the display active area. ie. Xposition + Xsize =< Xsrcsize

| Bit | Description |
|---|---|
| 31:28 | **Reserved** Project: All Format: MBZ |
| 27:16 | **Sprite_Y-Position** Project: All Format:<br>These 12 bits specify the vertical position in lines of the sprite (upper left corner) relative to the beginning of the active video area. When performing 180° rotation, this field specifies the vertical position of the lower right corner relative to the end of the active video area in the unrotated orientation. The defined sprite rectangle must always be completely contained within the displayable area of the screen image. |

| DVSBPOS—Video Sprite B Position Register | |
|---|---|
| 15:12 | **Reserved** Project: All Format: MBZ |
| 11:0 | **Sprite_X-Position** Project: All Format:<br>These 12 bits specify the horizontal position in pixels of the sprite (upper left corner) relative the beginning of the active video area. When performing 180° rotation, this field specifies the horizontal position of the original lower right corner relative to the original end of the active video area in the unrotated orientation. The defined sprite rectangle must always be completely contained within the displayable area of the screen image. |

## 4.11.5 DVSBSIZE—Video Sprite B Size Register

| DVSBSIZE—Video Sprite B Size Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **73190h-73193h** |
| **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 DVSBSURF |

This register specifies the size of the sprite. Software must take care that the sprite does not extend out of the display active area. ie. Xposition + Xsize =< Xsrcsize

| Bit | Description |
|---|---|
| 31:28 | **Reserved** Project: All Format: MBZ |
| 27:16 | **Sprite_Height** Project: All Format:<br>This register field is used to specify the height of the sprite in lines. The value in the register is the height minus one. The defined sprite rectangle must always be completely contained within the displayable area of the screen image. |
| 15:12 | **Reserved** Project: All Format: MBZ |
| 11:0 | **Sprite_Width** Project: All Format:<br>This register field is used to specify the width of the sprite in pixels. This does not have to be the same as the stride but should be less than or equal to the stride in pixels. The value in the register is the width minus one. The defined sprite rectangle must always be completely contained within the displayable area of the screen image.<br>The sprite width (actual width, not the width minus one value) is limited to even values when YUV source pixel format is used, or Pixel Multiply is set to Line/Pixel doubling or Pixel doubling. |

### 4.11.6 DVSBKEYVAL—Video Sprite B Color Key Value Register

| DVSBKEYVAL—Video Sprite B Color Key Value Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **73194h-73197h** |
| **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 key color to be used with the mask bits to determine if the sprite source data matches the key. This register will only have an effect when the sprite color key is enabled. In source key mode this value is the minimum value for the range compare. In destination key mode this value is the compare value.

| Bit | Description |
|---|---|
| 31:24 | **Reserved** Project: All Format: MBZ |
| 23:16 | **V_Source_Key_Min_Value/R_Source/Dest_Key_Value** Project: All Format:<br>Specifies the color key (minimum) value for the sprite V channel source key or the Red channel source or destination key compare value. |
| 15:8 | **Y_Source_Key_Min_Value/G_Source/Dest_Key_Value** Project: All Format:<br>Specifies the color key (minimum) value for the sprite Y channel source key or the Green channel source or destination key compare value. |
| 7:0 | **U_Source_Key_Min_Value/B_Source/Dest_Key_Value** Project: All Format:<br>Specifies the color key (minimum) value for the sprite U channel source key or the Blue channel source or destination key compare value. |

## 4.11.7 DVSBKEYMSK—Video Sprite B Color Key Mask Register

### DVSBKEYMSK—Video Sprite B Color Key Mask Register

| | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **73198h-7319Bh** |
| **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 range checking on.

For destination key this register specifies the key mask to be used with the color value bits to determine if the display source data matches the key when enabled. 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_Channel_Source_Key_Enable** Project: All Format:<br>Specifies the source color key enable for the V/Red channel |
| 25 | **Y/G_Channel_Source_Key_Enable** Project: All Format:<br>Specifies the source color key enable for the Y/Green channel |
| 24 | **U/B_Channel_Source_Key_Enable** Project: All Format:<br>Specifies the source color key enable for the U/Blue channel |
| 23:16 | **R_mask_Dest_Key_Value** Project: All Format:<br>Specifies the destination color key mask for the sprite R channel |
| 15:8 | **G_mask_Dest_Key_Value** Project: All Format:<br>Specifies the destination color key mask for the sprite G channel |
| 7:0 | **B_mask_Dest_Key_Value** Project: All Format:<br>Specifies the destination color key mask for the sprite B channel |

## 4.11.8 DVSBSURF—Video Sprite B Surface Address Register

| DVSBSURF—Video Sprite B Surface Address Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **7319Ch-7319Fh** |
| **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 DVSB registers** | |

| Bit | Description |
|---|---|
| 31:12 | **Sprite_Surface_Base_Address**<br>Project: All<br>Address: GraphicsAddress[31:12]<br>This address specifies the surface base address. When the surface is tiled, panning is specified using (x, y) offsets in the DVSBTILEOFF register. When the surface is in linear memory, panning is specified using a linear offset in the DVSBLINOFF register.<br>This address must be 4K aligned. It represents an offset from the graphics memory aperture base and is mapped to physical pages through the global GTT. The value in this register is updated through the command streamer with synchronous flips. |
| 11:0 | **Reserved** Project: All Format: |

## 4.11.9 DVSBKEYMAXVAL—Video Sprite B Color Key Max Value Register

| DVSBKEYMAXVAL—Video Sprite B Color Key Max Value Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **731A0h-731A3h** |
| **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 key color to be used with the mask bits to determine if the sprite source data matches the key. This register will only have an effect when the sprite source color key is enabled.

| Bit | Description |
|---|---|
| 31:24 | **Reserved** Project: All Format: MBZ |
| 23:16 | **V_Key_Max_Value** Project: All Format:<br>Specifies the color key value for the sprite V channel |

| DVSBKEYMAXVAL—Video Sprite B Color Key Max Value Register | | |
|---|---|---|
| 15:8 | **Y_Key_Max_Value** Project: All Format:<br>Specifies the color key value for the sprite Y channel | |
| 7:0 | **U_Key_Max_Value** Project: All Format:<br>Specifies the color key value for the sprite U channel | |

## 4.11.10DVSBTILEOFF—Video Sprite B Tiled Offset Register

| DVSBTILEOFF—Video Sprite B Tiled Offset Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **731A4h-731A7h** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | R/W |
| **Size (in bits):** | 32 |
| **Trusted Type:** | 1 |
| **Double Buffer Update Point:** | Start of vertical blank or pipe disabled |

This register specifies the panning for the sprite surface in tiled memory. The surface base address is specified in the DVSBSURFADDR register, and this register is used to describe an offset from that base address. When the surface is in linear memory, the offset is specified in the DVSBLINOFF register and the contents of this register are ignored. When the surface is tiled, the start position is specified in this register as an (x, y) offset from the beginning of the surface.

| Bit | Description |
|---|---|
| 31:28 | **Reserved** Project: All Format: MBZ |
| 27:16 | **Sprite_Start_Y-Position** Project: All Format:<br>These 12 bits specify the vertical position in lines of the beginning of the active display plane relative to the display surface. When performing 180° rotation, this field specifies the vertical position of the lower right corner relative to the start of the active display plane in the unrotated orientation. |
| 15:12 | **Reserved** Project: All Format: MBZ |
| 11:0 | **Sprite_Start_X-Position** Project: All Format:<br>These 12 bits specify the horizontal offset in pixels of the beginning of the active display plane relative to the display surface. The offset must be even pixel aligned for unrotated YUV formats. When performing 180° rotation, this field specifies the horizontal position of the lower right corner relative to the start of the active display plane in the unrotated orientation. |

## 4.11.11DVSBSURFLIVE—Video Sprite B Live Surface Base Address Register

| DVSBSURFLIVE—Video Sprite B Live Surface Base Address Register | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **731ACh-731AFh** |
| **Project:** | All |
| **Default Value:** | 00000000h |
| **Access:** | Read Only |
| **Size (in bits):** | 32 |

| Bit | Description |
|---|---|
| 31:0 | **Sprite_Surface_Base_Address**<br>Project: All<br>Address: GraphicsAddress[31:0]<br>This gives the live value of the surface base address as being currently used for the plane. |

## 4.11.12DVSBSCALE—Video Sprite B Scaler Control

| DVSBSCALE—Video Sprite B Scaler Control | |
|---|---|
| **Register Type:** | MMIO |
| **Address Offset:** | **73204h-73207h** |
| **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 DVSBSURF |

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

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. Scaling can not be used with the sprite 64bpp source pixel format. Source and destination sizes must be 3x3 (3x6 when interlacing) or greater when scaling is enabled.

Horizontal downscaling limits the maximum dot clock allowed as percent of cdclk. See DVSASCALE for the rules to calculate the allowed dot clock.

| Bit | Description |
|---|---|
| 31 | **Scaling_Enable** Project: All Format: Enable<br>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. |

## DVSBSCALE—Video Sprite B Scaler Control

30:29 **Filter_Control**

Project: All

Default Value: 0b

Filter selection

| Value | Name | Description | Project |
|---|---|---|---|
| 00b | Medium | Medium Filtering | All |
| 01b | Enhancing | Edge Enhancing Filtering | All |
| 10b | Softening | Edge Softening Filtering | All |
| 11b | Reserved | Reserved | All |

28 **Even/Odd_Field_Offset**

Project: All

Default Value: 0b

Select the vertical offset of the filtered data. Software is responsible for updating this to match the surface data.

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | 0 | Vertical initial phase of 0 | All |
| 1b | 0.5 | Vertical initial phase of 0.5 | All |

27 **Even/Odd_Field_Enable**

Project: All

Default Value: 0b

Enable adjustment of the vertical offset of the filtered data.

| Value | Name | Description | Project |
|---|---|---|---|
| 0b | Disable | Off (Vertical initial phase is 1/2 the scale factor) | All |
| 1b | Enable | On (Vertical initial phase is selected by the Even/Off Field Offset bit) | All |

26:16 **Source_Width** 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 source pixel format is used.

15:11 **Reserved** Project: All Format: MBZ

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.

## 4.11.13 DVSBGAMC—Video Sprite B Gamma Correction Registers

### DVSBGAMC—Video Sprite B Gamma Correction Registers

**Register Type:** MMIO
**Address Offset:** 73300h
**Project:** All
**Default Value:** 00000000h; 04010040h; 08020080h; 0C0300C0h; 10040100h; 14050140h; 18060180h; 1C0701C0h; 20080200h; 24090240h; 280A0280h; 2C0B02C0h; 300C0300h; 340D0340h; 380E0380h; 3C0F03C0h; 00000400h; 00000400h; 00000400h;
**Access:** R/W
**Size (in bits):** 19x32

See Video Sprite A description

| DWord | Bit | Description |
|---|---|---|
| 0 | 31:0 | **GAMC0** Project: All Format: DVSGAMC Reference Point |
| 1 | 31:0 | **GAMC1** Project: All Format: DVSGAMC Reference Point |
| 2 | 31:0 | **GAMC2** Project: All Format: DVSGAMC Reference Point |
| 3 | 31:0 | **GAMC3** Project: All Format: DVSGAMC Reference Point |
| 4 | 31:0 | **GAMC4** Project: All Format: DVSGAMC Reference Point |
| 5 | 31:0 | **GAMC5** Project: All Format: DVSGAMC Reference Point |
| 6 | 31:0 | **GAMC6** Project: All Format: DVSGAMC Reference Point |
| 7 | 31:0 | **GAMC7** Project: All Format: DVSGAMC Reference Point |
| 8 | 31:0 | **GAMC8** Project: All Format: DVSGAMC Reference Point |
| 9 | 31:0 | **GAMC9** Project: All Format: DVSGAMC Reference Point |
| 10 | 31:0 | **GAMC10** Project: All Format: DVSGAMC Reference Point |
| 11 | 31:0 | **GAMC11** Project: All Format: DVSGAMC Reference Point |
| 12 | 31:0 | **GAMC12** Project: All Format: DVSGAMC Reference Point |
| 13 | 31:0 | **GAMC13** Project: All Format: DVSGAMC Reference Point |
| 14 | 31:0 | **GAMC14** Project: All Format: DVSGAMC Reference Point |
| 15 | 31:0 | **GAMC15** Project: All Format: DVSGAMC Reference Point |
| 16 | 31:0 | **GAMCmaxR** Project: All Format: DVSGAMC Max Reference Point |
| 17 | 31:0 | **GAMCmaxG** Project: All Format: DVSGAMC Max Reference Point |
| 18 | 31:0 | **GAMCmaxB** Project: All Format: DVSGAMC Max Reference Point |

# Revision History

| Revision Number | Description | Revision Date |
|---|---|---|
| 1.0 | First 2011 OpenSource edition | May 2011 |
| | | |

§§